

Arbeitsteilung zwischen direkter Manipulation und proaktiven Software-Agenten in agentengestützten Informationssystemen

Von der Fakultät für Ingenieurwissenschaften

der Universität Duisburg-Essen

zur Erlangung des akademischen Grades eines

Doktors (Dr. rer. nat.)

genehmigte Dissertation

von

André Schaefer

aus Düsseldorf

Referent: Prof. Dr. Norbert Fuhr

Korreferent: Prof. Dr. Jürgen Krause

Tag der mündlichen Prüfung: 5. 6. 2008

Inhaltsverzeichnis

1	Einleitung	7
2	Information Retrieval in verteilten digitalen Bibliotheken	11
2.1	Iterative Natur des Information Retrieval	14
2.2	Komplexität und Orientierung	14
2.3	Verteilte Suche	18
2.4	Heterogenität	18
2.5	Direkte Manipulation und Visualisierung in Informationssystemen . .	20
2.5.1	Formulare	22
2.5.2	Menüsteuerung	23
2.5.3	Ikonen, Maus und Zeiger	24
2.5.4	Visuelle Formalismen	25
2.5.5	Graphen	26
2.5.6	Visualisierung	28
2.5.7	Spezifika für das Textretrieval	31
2.6	Agenten	34
2.6.1	Mehrwertdienste	34
2.6.2	Strategische Unterstützung	34
2.6.3	Monitoring und Awareness	35
2.6.4	Integration, Homogenisierung und Filtern	35
2.7	Gemeinsame Anwendung	35
3	Arbeitsteilung in agentengestützten Informationssystemen	37
3.1	Proaktivität versus direkte Manipulation	37
3.1.1	Direkte Manipulation versus Software-Agenten	38
3.1.2	Benutzungsschnittstellen mit gemischter Initiative	40
3.1.3	Agenten und proaktive, intelligente Systemkomponenten . . .	43
3.1.3.1	Definitionen	43

Inhaltsverzeichnis

3.2	Mentale und kognitive Modelle	46
3.2.1	Metaphern	46
3.2.2	Kooperationsmodelle	48
3.2.3	Psychologische Aspekte	48
3.3	Einsatz von Agenten in Informationssystemen	55
3.3.1	Information Agents	55
3.3.1.1	Information Broker	55
3.3.1.2	Monitoring Agents	58
3.3.1.3	Just in Time Information Retrieval Agents	58
3.3.2	Interface Agents	61
3.3.2.1	Benutzungsschnittstellen	62
3.3.2.2	Avatare, Personas	70
3.3.2.3	Personalisierungsagenten	71
3.4	Entwurfsmuster	72
3.4.1	Just In Time Information Retrieval	72
3.4.2	Eager Assistants	72
3.4.3	Wizards	73
3.4.4	Status-Manager	74
3.4.5	Aktive Hilfe	74
3.4.6	Path Marker	75
3.5	Strategische Unterstützung durch Agenten in digitalen Bibliotheken	76
3.5.1	Unterstützung durch hochwertige Suchfunktion	80
3.5.2	Föderierte Suche	80
3.5.3	Integrierte Suche	80
3.5.4	Schalenmodell	81
3.5.5	Informationeller Mehrwert	82
4	Daffodil	83
4.1	Ausgangspunkt und Zielsetzung	83
4.2	Projektbeschreibung	84
4.2.1	Projektverlauf	85
4.2.2	Modellierung der Rechercheoperationen	85
4.3	WOB Modell	87
4.3.1	Generelle softwareergonomische Prinzipien	88
4.3.2	Iteratives Retrieval	89

Inhaltsverzeichnis

4.3.2.1	Modalitätsmischung	90
4.3.2.2	Zustandsanzeige mit Korrekturmodus	91
4.3.3	Strikte Objektorientierung	91
4.3.4	Werkzeugmetapher und doppelte Interpretierbarkeit	92
4.4	Architektur, Agenten und Dienste	93
4.4.1	Modularität	95
4.4.2	Nachrichtenarchitektur	96
4.4.3	Interaktionsschemata	97
4.5	Aufgaben und Szenarien für Agenten im Daffodil Kontext	98
4.5.1	Föderierte Suche	98
4.5.2	Agentengestützte Personalisierung und Adaptivität	100
4.5.3	Eingabeunterstützung und Anfrageunterstützung	104
4.5.4	Unterstützung der iterativen Anfrage und Resultatauswertung	109
4.5.5	Aufbau und Pflege der persönlichen Handbibliothek	113
4.5.5.1	Awareness	113
4.5.5.2	Monitoring	114
4.5.5.3	Recommendation	114
4.6	Benutzungsoberfläche	114
4.6.1	Anforderungen	115
4.6.2	Graphische Oberfläche und ihre Werkzeuge	117
4.6.2.1	Suche	117
4.6.2.2	Persönliche Handbibliothek	119
4.6.2.3	Referenzen und Zitationen	122
4.6.2.4	Autorennetzwerke	123
4.6.2.5	Journale und Konferenzen	125
4.7	Fazit Daffodil	125
5	Evaluation	127
5.1	Daffodil Phase I	129
5.1.1	Ziele	129
5.1.2	Umfang und Teilnehmer	130
5.2	Daffodil Phase II	131
5.2.1	Strategische Unterstützung	131
5.2.1.1	Ziele der Evaluation	131
5.2.1.2	Durchführung der Evaluation und Resultate	132

Inhaltsverzeichnis

5.2.2	Evaluation der proaktiven Vorlage bei der Anfrageformulierung	136
5.2.2.1	Ziele der Evaluation	136
5.2.2.2	Durchführung und Resultate	136
5.2.2.3	Heuristische Evaluation mit Mock-Ups	137
5.2.2.4	Ergebnisse der heuristischen Evaluation	138
5.2.2.5	Kontrollierte Benutzertests in zwei Gruppen	139
5.2.2.6	Benutzertests mit Videoprotokoll	142
5.2.2.7	Zufriedenheit der Testpersonen	145
6	Fazit	147
A	Evaluation	150
A.1	Der Fragebogen für die Evaluation der Baseline	150
A.1.1	Vor jeder Aufgabe	150
A.1.2	Nach jeder Aufgabe	150
A.2	Der Fragebogen für die Evaluation der proaktiven Version von Daffodil	150
A.2.1	Vor jeder Aufgabe	150
A.2.2	Nach jeder Aufgabe	151
A.2.3	Am Ende der Sitzung	151
B	Kurzübersicht Klassenbibliothek	153
B.1	WOB-Werkzeuge und Desktop	153
B.1.1	Überblick	153
B.1.2	Anforderungen	154
B.1.3	WOB-Bibliothek	154
B.2	Klassenmodell	156
B.2.1	Besonderheiten	156
B.3	Anwendungskontexte	157
B.3.1	Neue Tools definieren	157
B.3.1.1	Initialisierung	158
B.3.1.2	View erzeugen und initialisieren	159
B.3.2	Neue Tools anmelden	160
B.4	Event-Modell	162
B.4.1	Kommunikation	162
B.4.2	Kommunikationsvarianten	162
B.4.2.1	Bearbeitungsreihenfolge	164

Inhaltsverzeichnis

B.4.2.2	EventReciever	165
---------	-------------------------	-----

Literaturverzeichnis	169
-----------------------------	------------

Danksagung

Diese Arbeit wäre ohne die direkte und indirekte Unterstützung der folgenden Personen nicht entstanden.

Ich bedanke mich bei Professor Dr. Norbert Fuhr, Professor der Informatik an der Universität Duisburg-Essen, der mir viel Unterstützung zukommen lassen hat. Ohne ihn hätte ich mein Promotionsvorhaben nicht verfolgen können. Ebenso gebührt mein Dank Professor Dr. Jürgen Krause, Professor für Computervisualistik an der Universität Koblenz-Landau und wissenschaftlicher Direktor des Informationszentrums Sozialwissenschaften, Bonn, der mir viele Grundlagen vermittelt und die Themenfindung ermöglicht hat. Sein Beitrag ist das Fundament dieser Arbeit.

Ich bedanke mich auch bei meinen jeweiligen Kollegen für regen Austausch und Ideen, ohne die konkrete Ergebnisse nicht erreichbar gewesen wären: Thomas Mandl, Maximilian Stempfhuber, Maximilian Eibl, Peter Mutschke, Markus Schommler, Udo Riege, Robert Strötgen, Bernd Hermes, Riehli Neitzke, Gudrun Fischer, Henrik Nottelmann, Kai Grossjohann, Andrea Ernst-Gerlach, Ingo Frommholz, Sascha Kriewel und Saadia Malik. Besonderen Dank möchte ich meinem langjährigen Projektpartner Peter Klas sagen, der vieles praktisch möglich gemacht hat, was sonst nur graue Theorie geblieben wäre. Allen weiteren Beteiligten, die ich nicht namentlich aufzählen konnte, an dieser Stelle vielen Dank.

Bei der Fehlersuche und fachlichen Korrektur haben Keywan Najafi Tonekaboni und Matthias Jordan wertvolle Beiträge geliefert. Herzlichen Dank an Valentin Richter, für Motivation und Unterstützung.

Schließlich möchte ich mich bei meiner Frau Isabell und meiner ganzen Familie und meinen Freunden für ihre Geduld und tatkräftige Unterstützung danken.

1 Einleitung

Wissenschaftliche Information verteilt sich auf viele unterschiedliche Quellen. Für umfassende Recherchen sind die Wissenschaftler auf verteilte Angebote und Informationsquellen angewiesen und müssen ihre Anfragen auf mehrere Quellen verteilen. Um die steigende Informationsflut und Arbeitsbelastung bei der Recherche zu bewältigen, brauchen Anwender zunehmend Hilfe, um gezielt die richtigen Informationsquellen zu finden und Information zu *sammeln*, *verknüpfen* und zu *strukturieren*.

Der Informationsraum für die wissenschaftliche Recherche wird zunehmend größer und heterogener. Es wird für den einzelnen Wissenschaftler zwar auch zunehmend leichter, auf digitalem Weg an eine Veröffentlichung zu kommen, jedoch gibt es keinen zentralen Zugangspunkt, der die Aufgaben einer Fachbibliothek im digitalen Zeitalter übernimmt. Virtuelle Fachbibliotheken und zusammenführende Meta-Informationssysteme sind zwar in der Entstehung, jedoch decken sie meist, auch aus rechtlichen Gründen, nicht das ganze Angebot ab. Ein andauernder Streit über die Rechte an den Veröffentlichungen zwischen Verlagen und Autoren zementiert die bestehende Heterogenität auf absehbare Zeit.

Eine Hauptaufgabe für das Recherchieren in digitalen Bibliotheken ist also das Management der Heterogenität und Vielfalt. Dass für diese Aufgaben zunehmend Softwareagenten benutzt werden, liegt an deren Vorzügen in Hinblick auf verteilte Systeme sowie ihre *Autonomie* und ihre Flexibilität bei der Anpassung an sich verändernde Umgebungen.

- Sie sammeln und indexieren Informationen in Computernetzen und Dateisystemen.
- Sie lernen und bilden Wissensbasen im Dienste der Anwender.
- Sie kennen und verfeinern Benutzerprofile, um ihre Ergebnisse an den Anwenderkontext und die Vorlieben der Anwender anzupassen.

1 Einleitung

Trotz dieser Vorzüge gibt es reichlich Probleme in Hinblick auf die Dialoggestaltung, da die Erfordernisse oftmals im Konflikt mit dem vorherrschenden Paradigma der *Benutzerautonomie* und der *direkten Manipulation* stehen. Zudem ist gerade die Recherche nach Informationen eng verknüpft mit Unsicherheit auf Seiten der Benutzer, die sich in der Regel in einem Problemlösungsprozess befinden. Dieser bedingt mentale Zustände, die besonders störanfällig und fragil sind. Die Arbeitsabläufe sind ständigen Richtungsänderungen unter starkem Einfluss von Rückkopplung unterworfen. Bis der Informationssuchende ein zufriedenstellendes Ergebnis erzielt hat, hat er viele Suchpfade beschritten und hat iterativ sein Informationsbedürfnis strukturiert, konturiert und verfeinert. Er hat explorativ gearbeitet, ist assoziativ vorgegangen und hat vielerlei Informationen gesammelt und verknüpft, bis sich die Information zu Wissen verdichtet hat. Seine mentalen Modelle entwickeln sich dabei kontinuierlich.

Um zum Ziel zu gelangen, muss der Suchende jedoch ein nennenswertes Maß an Informationskompetenz erlangen. Er muss wissen, wo und wie er an die benötigten Informationen kommt und wie er diese geschickt auswertet, um weitere Ansatzpunkte zu finden. Auch muss er durch viel Erfahrung gehöriges strategisches und taktisches Geschick erlangen. Viele Anwender sind jedoch in dieser Situation überfordert und brauchen dringend Unterstützung. In der Vergangenheit war diese Unterstützung bei professionellen Informationsvermittlern und Bibliothekaren zu finden. Bei der nun allgegenwärtig gewordenen Rechercheproblematik im Internet ist so ein Experte jedoch nicht in jedem Fall verfügbar. Die Softwareindustrie hofft, die zunehmend auf sich gestellten Anwender mit Software-Agenten zu unterstützen, um die Lücke zu füllen. Es ist klar, dass Software-Agenten in diesem Szenario eine schwierige Aufgabe haben: Sie müssen den fragilen Recherche-Prozess unterstützen, ohne ihn in die falsche Richtung zu steuern. Sie müssen die Autonomie des Benutzers berücksichtigen, ihn aber auch davor bewahren, sich im Kreis zu drehen, falschen Fährten zu folgen oder sich zu verzetteln. Vielleicht benötigt der Recherchierende besondere Unterstützung in Hinblick auf seine Strategien, um effizienter und effektiver zu recherchieren. Viele Hinweise auf die richtigen Pfade können helfen, im ungünstigen Fall aber ebenso verwirren.

Andere Ansätze, dieses Dilemma aus steigendem Informationsangebot und tendenziell geringerer Informationskompetenz zu bewältigen, sind vor allem im Bereich der Informationsvisualisierung entstanden. Dieser Ansatz steht nicht im Widerspruch zur Benutzerschnittstellen-Metapher der direkten Manipulation. Daher wird er von Vertretern dieses Paradigmas als Alternative zu den Software-Agenten gesehen, oh-

1 Einleitung

ne die Benutzer-Autonomie und die Vorhersagbarkeit und Steuerbarkeit der Benutzungsoberfläche in Frage zu stellen (Shneiderman u. Maes 1997; Shneiderman 1997). Shneiderman sieht auch sein Konzept der *dynamic queries*, bei denen der Anwender durch die Nutzung von Kontrollelementen sehr schnell durch große Datenmengen navigieren, als möglichen Ausweg aus dem Recherche-Dilemma (Shneiderman 1994). Zu den Ansätzen der Rechercheoberflächen im Information Retrieval siehe Abschnitt 2.5.

Bei der Frage der Gestaltung der Interaktion mit Software-Agenten oder in Software-Systemen, in denen Software-Agenten eine Rolle spielen, muss über die passende Metaphorik entschieden werden: Welche Metaphern passen zu Software-Agenten? Müssen diese in jedem Fall direkt repräsentiert werden, oder können sie im Hintergrund wirken; wie wirkt sich das auf die mentalen Modelle der Anwender aus?

Maes sieht Software-Agenten eher als komplementäre Metapher zur direkten Manipulation oder zur Visualisierung. Auch beim Einsatz von Software-Agenten ist typischerweise eine sehr gute direkt-manipulative Benutzungsoberfläche notwendig. Die Software-Agenten können dem Benutzer auch gleichsam „über die Schulter schauen“ und ohne eigene visuelle Metapher auf die Software wirken (Shneiderman u. Maes 1997, S. 53).

Bei der Frage nach Automatisierung ist es wichtig, das richtige Maß zu finden:

- Welche arbeitsintensiven Schritte können automatisiert werden?
- Wann wird Automatisierung im Hinblick auf das Arbeitsziel der Anwender kontraproduktiv?
- Wie kann man das Delegieren von Aufgaben gestalten?

Zu der Paradigmen-Debatte siehe auch Abschnitt 3.1.1.

Software-Agenten wurden bereits in Informationssystemen verwendet. Dies war meist auf einen bestimmten Aspekt konzentriert, wie beispielsweise *kollaboratives Filtern*, *aktive Hilfe* oder *adaptive Konfiguration*. Die offene Wissensbasis bei föderierten digitalen Bibliotheken, bei denen die Daten nicht direkt vorliegen, sondern die Suchanfrage an eigenständige Datenquellen und digitale Bibliotheken verteilt wird, fordert jedoch eine konsequentere Umsetzung der sich bietenden Möglichkeiten. Insbesondere müssen diverse Bedienkonzepte zusammengeführt werden, um für den Anwender ein schlüssiges und kognitiv leicht zu erfassendes Ganzes bieten zu können. Nur so kann die Informationssuche und Verarbeitung tatsächlich effizienter werden. Die

1 Einleitung

gemischte Initiative von Benutzer und System soll auf ein gemeinsames, aber durch den Kontext des Benutzers vorgegebenes, Ziel ausgerichtet bleiben.

Ausgehend von einer systematischen Analyse der Arbeitsprozesse bei der wissenschaftlichen Informationssuche sowie von agentengestützten Informationssystemen als Softwaregattung und Gestaltungsmodellen für Informationssysteme wird ein Kriterienkatalog erarbeitet, der die zahlreichen Zielkonflikte benennt, berücksichtigt und ausgewogene Lösungsvorschläge anbietet. So kann bei der Gestaltung von agentengestützten Literaturrechersystemen zielgerichtet eine ergonomische Gesamtgestaltung angestrebt werden, die zu höherer Zufriedenheit, Effizienz und Effektivität und zur Bewältigung von Informationsflut und Arbeitslast beitragen kann.

Im Folgenden wird die Gestaltung von Informationssystemen zur wissenschaftlichen Literaturrecherche unter Verwendung von Software-Agenten in föderierten digitalen Bibliotheken betrachtet. Insbesondere wird die Frage behandelt, wie die Proaktivität der Agentenfunktionen gestaltet werden kann, so dass der Anwender sich unterstützt und nicht bevormundet, abgelenkt oder gestört fühlt.

Als Bezugspunkt für die Konzepte zu ergonomisch sinnvoller Aufgabenteilung dient die föderierte, virtuelle digitale Bibliothek DAFFODIL. DAFFODIL nutzt Agenten als Grundlage für die Zusammenführung von verschiedenen existierenden digitalen Bibliotheken und Informationsdiensten für den Wissensbereich der Informatik. Die Gewinnung von Mehrwertinformation zur strategischen Unterstützung der Anwender bei der Recherche ist das vorrangige Ziel. DAFFODIL bietet Ansatzpunkte für die Arbeitsteilung zwischen Anwender und Informationssystem auf verschiedenen Ebenen in Frontend (GUI) sowie Backend (Datenquellenanbindung, Ergebniszusammenführung, Objektverwaltung).

2 Information Retrieval in verteilten digitalen Bibliotheken

Die computergestützte Suche und Wiedergewinnung wissenschaftlicher Information in digitalen Bibliotheken (*Information Retrieval*) hat einen hohen Stellenwert erlangt. Information ist dort sehr aktuell, schnell verfügbar und vom Arbeitsplatz aus erreichbar. Die Dokumente sind durch ihre digitale Form gut verwendbar. Man kann mit Hilfe fortgeschrittener Methoden nach ihnen suchen und man kann die benötigten Informationen besser auf dem aktuellen Stand halten. Information ist so immer und überall verfügbar.

Arms beschreibt in „Digital Libraries“ Arms (2001), dass das Internet und das World Wide Web wichtige Grundbausteine dafür sind und dass Wissenschaftler sich zunehmend auf digitale Informationsquellen stützen und verlassen. Dennoch markieren die heute existierenden digitalen Bibliotheken vor allem den Übergang zwischen den traditionellen Bibliotheken (und der bisherigen Informationsversorgung von Wissenschaftlern durch Bücher und Artikeln in Journalen) hin zu einer noch offenen und unentschiedenen Zukunft, bei der diese Information umfassend digital zur Verfügung steht.

Im letzten halben Jahrhundert hat sich viel verändert, um dieses Ziel zu erreichen. Vannevar Bush beklagt in seinem Artikel „As we may think“ (Bush 1949), der in diesem Zusammenhang oft als grundlegende visionäre Schrift zitiert wird, dass die Menschheit zu dem gesammelten Wissen nur unzureichenden Zugang habe. Obwohl immer mehr Wissen angesammelt werde, wird diese Sammlung immer unzugänglicher, weil die Methoden zum Auffinden von Wissen nicht adäquat entwickelt seien. So gesehen, sei die Menschheit schlechter dran als zuvor, solange es nicht gelänge die gesammelte Information wieder zu erschließen.

Es wurden enorme Fortschritte durch die Entwicklung von Computern, elektronischer Information, Vernetzung und das Internet erreicht. Das Internet und insbeson-

dere das World Wide Web (WWW) führen zu Verknüpfungen der Informationsbestandteile und zur breiten Erreichbarkeit und Verfügbarkeit von Wissen. Das sei den Eigenschaften der menschlichen Kognition nach essentiell, wie Bush meint, der eine Art von assoziativer Indexierung wünschenswert findet, bei der jede Auswahl von Information dazu führen kann, dass automatisch und unmittelbar die folgende herausgesucht wird. Diese Art von Assoziativität wohnt einerseits den Hypertextmedien wie dem WWW inne, aber ebenso den Informationssystemen, die Mechanismen wie beispielsweise kollaboratives Filtern einsetzen.

Dennoch wurde Bushs Vision längst nicht vollständig erreicht. Arms (2001, p. 143) gibt zu bedenken, dass die hohe Rate an Veränderung einer der vorrangigen Nachteile der digitalen gegenüber der traditionellen Art von Bibliothek sei. Während die traditionelle Bibliothek sich kaum verändere, sei Veränderung bei digitalen Bibliotheken ein ständiges Thema. Jeder Monat bringe neue Services, neue Kollektionen, neue Schnittstellen und „neue Kopfschmerzen“. Die Benutzbarkeit leide unter den ständigen Neuerungen, weil Anwender nicht so schnell und oft Basiswissen wieder erlernen möchten. Der ständige Wandel scheint ein notwendiges Übel von digitalen Bibliotheken zu sein.

Arms weist auch darauf hin, dass Systeme mit fortschrittlicher Technik oft von Experten für Experten entworfen und gebaut werden. Wenn ungeschulte Nutzer über das Internet auf Systeme zugreifen können, die Training oder tiefer gehendes Wissen oder auch technischen Support nötig machen, sind sie oft frustriert. Sie wollen ihre Zeit nicht für solche Hürden opfern. Öffentlich zugängliche Informationssysteme sollten daher mit minimalem Training bedienbar sein.

Bei der großen Menge an stetig neuer Information ist die Bewertung und Bewertbarkeit der Qualität der jeweiligen Information für den Anwender ein herausragend wichtigen Kriterium. Das Bewusstsein dazu stellt sich in der Wissenschaft und den Anwendern jedoch erst in jüngerer Zeit ein. Eine Analyse der derzeitigen Möglichkeiten und Verfahren, insbesondere in Hinblick auf die speziellen Eigenschaften von Internetinhalten liefert (Mandl 2006). Mandl geht insbesondere auch für diesen Aspekt auf die mögliche Arbeitsteilung zwischen menschlichen, subjektiven Qualitätsurteilen, sowie berechenbaren, automatischen Qualitätsbewertungsverfahren, durch statistische und strukturelle Analysen ein.

Viele Bereiche bleiben jedoch für die Crawler und Spider allgemeiner Suchmaschinen unzugänglich, weil diese Informationen nicht direkt verlinkt sind, sondern über Da-

tenbankschnittstellen nur auf spezifische Anfragen dynamisch angeboten werden - das ist das sogenannte „Invisible Web“ Problem (Sherman u. Price 2003) beziehungsweise das „Deep Web“ Problem. Es ist für qualitativ hochwertige digitale Bibliotheken besonders spürbar, weil gerade im Bereich der wissenschaftlichen Publikationen ein großer Teil der verfügbaren Information in Datenbanken verwaltet wird (Lewandowski u. Mayr 2006). Viele Bereiche sind absichtlich vor dem Zugriff durch Suchmaschinen geschützt, um nur bestimmten Benutzergruppen Zugang zu gewähren oder um die Abrechnung kostenpflichtiger Angebote zu ermöglichen. Diese Variante des Deep Web wird in der Kategorisierung von Sherman u. Price (2001) „Proprietary Web“ genannt. Es gibt Schätzungen, die davon ausgehen, dass der Inhalt des Deep Web 40 bis 500 mal größer sein kann als der für Suchmaschinen erreichbare Inhalt, das sogenannte *Surface Web* (Bergman 2001).

Die verfügbare Information, speziell im wissenschaftlich relevanten Bereich ist auch längst nicht so gut verknüpft, wie es wünschenswert wäre. Für viele Daten finden sich relevante zugehörige Informationen zwar im World Wide Web wieder, müssen aber mühsam gesammelt werden. Findet sich etwa ein relevantes Dokument im Informationssystem eines Verlages, so ist längst nicht sofort ersichtlich, wo der Autor arbeitet, welche URL seine Homepage hat, in welchen Kontexten das Dokument zitiert wird, usw. Manche Informationssysteme bieten einige dieser Links an, aber bleiben dabei meist in einem klar abgesteckten Bereich, etwa der der „eigenen“ Publikationen und bieten so nur eine Teillösung an. (Krause 2007) thematisiert die Möglichkeiten der Verbindung solcher Informationsbestände mit den Mitteln des *semantic Web*, also durch semantische Abbildung auf der Basis von Ontologien, beziehungsweise durch semantische Transferkomponenten. Es wird möglich auf der Basis verteilt gepflegter und erschlossener hochwertiger Information integrierte Informationsangebote anzubieten.

Die folgenden Abschnitte befassen sich mit den Hauptaspekten der agentengestützten, föderierten Informationssysteme: Der verteilten Suche, der Heterogenitätsbehandlung, dem Problem von Komplexität und Orientierung sowie den Software-Agenten, als wichtige Maßnahme zur Bewältigung der genannten Probleme. Außerdem wird abschließend ein kurzer Ausblick auf die Besonderheit der gemeinsamen Anwendung dieser Konzepte gegeben, welcher im Verlauf dieser Arbeit am Beispiel des Informationssystems DAFFODIL dargestellt wird.

2.1 Iterative Natur des Information Retrieval

Information Retrieval ist eine iterative Tätigkeit. Eine Recherche besteht also nicht aus einem oder mehreren seriellen Schritten, sondern bestimmte Schritte müssen in Schleifen immer wieder mit leichten Variationen durchgeführt werden.

Die Recherchetätigkeit gliedert sich dabei bei den meisten Modellierungen in Anfragespezifikation, Erhalten einer Ergebnisliste und anschließender Anfragemodifikation, so lange bis ein befriedigendes Ergebnis zustande gekommen ist. Nach Ingwersen (1992) ergibt sich das Verhaltensmodell in Abbildung 2.1. Ingwersen berücksichtigt auch die Modifikation der Quellenauswahl und der Strategiewahl. Er nimmt auch an, dass die Interaktion mit einem menschlichen Informationsvermittler oder einem Informationssystem abwechseln kann. Zudem wird die *Anomalous State of Knowledge* These (ASK) von Belkin (1980) berücksichtigt, was aufgrund der implizierten Unsicherheit auf Seiten des Informationssuchenden wichtig ist, wie später noch ausgeführt wird¹.

Die Handlungen des Benutzers bei der Recherche sind meist nicht strikt zielgerichtet, sondern folgen einer „Berry picking“ Strategie, wie sie von (Bates 1989) benannt wurde. Dabei folgt die sich auf dem Weg zur Lösung entwickelnde Strategie unterschiedlichen taktischen Richtungen, bis sie letztlich zum Ziel gelangt, wie in Abbildung 2.2 zu sehen ist.

Entscheidend dabei ist, dass die Informationssuchenden bei der Recherche dazulernen und sich auch über ihre anfangs diffusen Ziele klarer werden. Zudem finden sie bei der Recherche neue Quellen oder wechseln ihre Taktik, um beispielsweise aus einer Sackgasse herauszukommen.

2.2 Komplexität und Orientierung

Der Informationssuchende, der in digitalen Bibliotheken recherchiert, sieht sich keinem geschlossenen Informationssystem mit festem Datenbestand gegenüber, sondern in der Regel einer *offenen Welt*, die ständig erweiterbar bleibt und deren Grenzen

¹Wie (Hearst 1999, S. 263) ausführt, legen die meisten Suchmaschinen ein vereinfachtes Modell zu Grunde, das annimmt, das Informationsbedürfnis sei während der Recherche statisch und präsentieren lange unorganisierte Resultatlisten – was bei vielen Anwendern auf wenig Gegenliebe stößt.

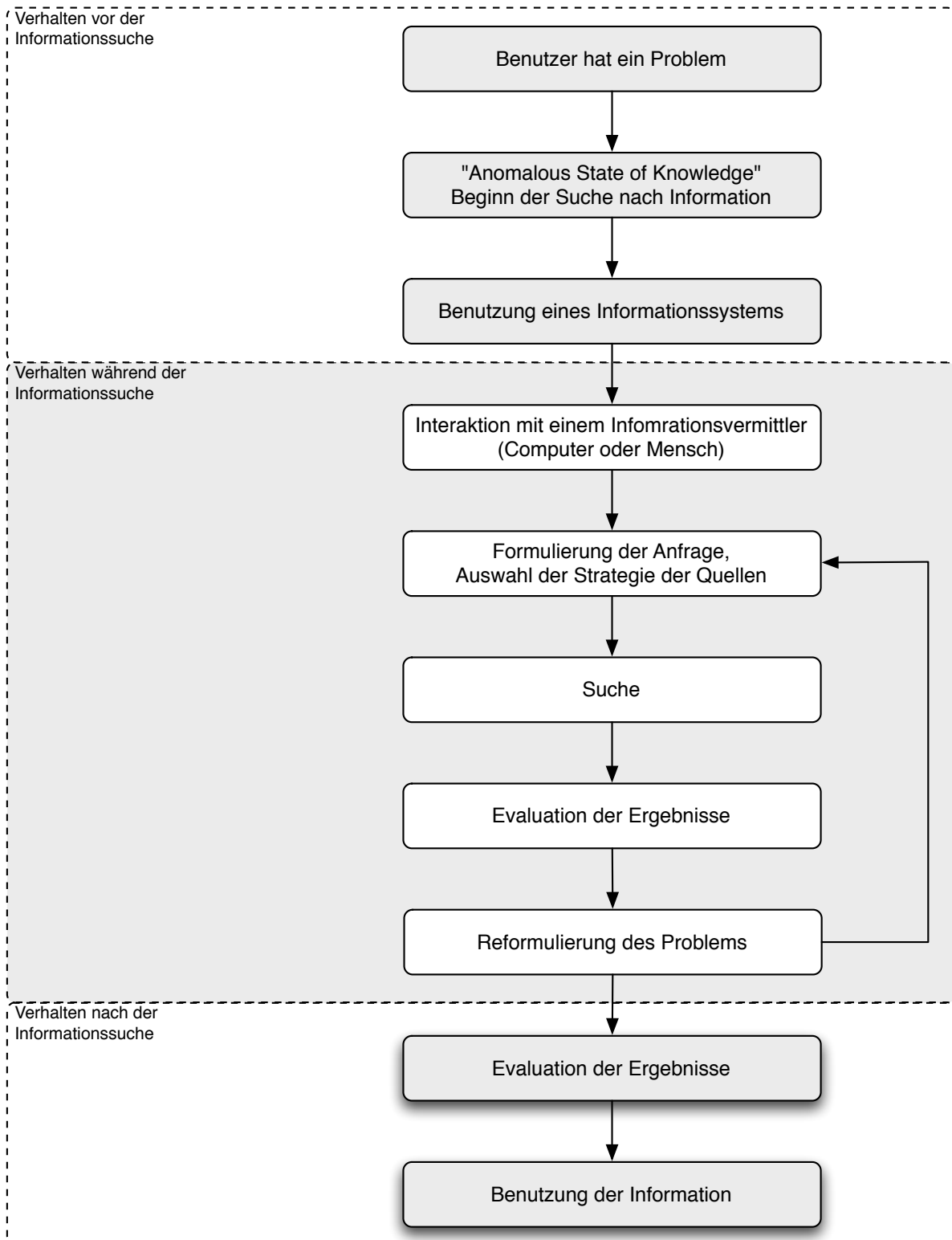


Abbildung 2.1: Verhalten bei der Informationssuche nach Ingwersen (1992)

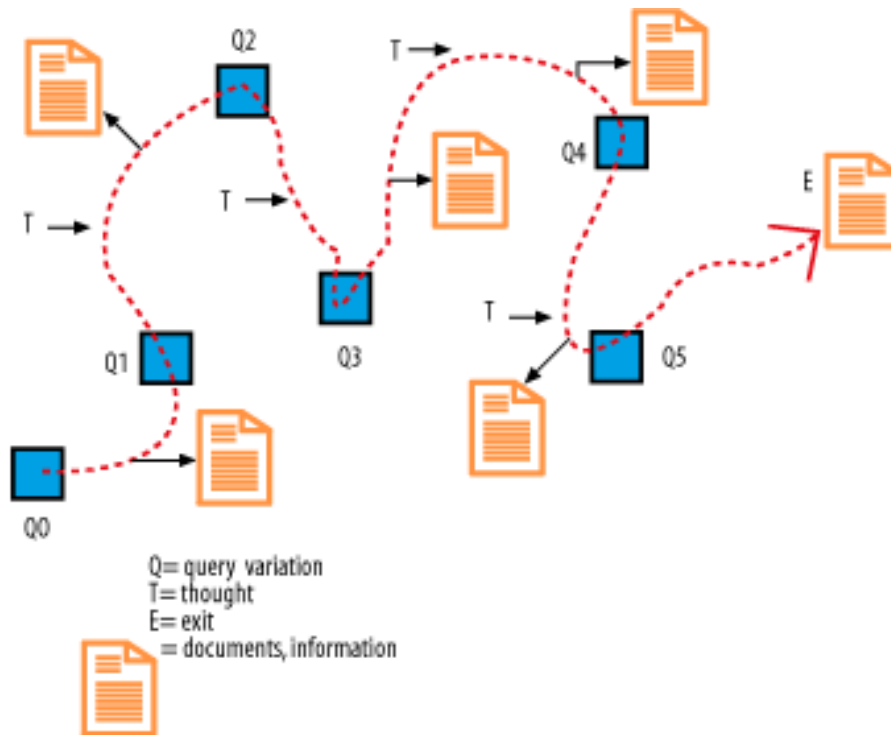


Abbildung 2.2: Berrypicking: Eine wechselhafte Folge von Rechenschritten

sich dauernd verändern. Eine wirklich umfassende Suche nach Information ist deshalb prinzipiell kaum oder nur sehr aufwendig möglich. Das von Arms beschriebene Dilemma (s.o.) gilt also noch.

Das Rechercheproblem ist komplex. Es kann zum Beispiel leicht sein, dass sich durch das Browsen von Information, dem Folgen von Links oder der Recherche keine *Orientierung* einstellt, was wie zusammenhängt – oder wie man zu einem bestimmten Punkt der Recherche gelangt ist.

Im Vergleich zu den oft gut strukturierten Sammlungen nicht-digitaler Bibliotheken, deren Örtlichkeit der Orientierung entgegenkommt, erscheinen digitale Informationen oft durch mangelnde Aufbereitung als eine große Zahl von Einzelinformationen ohne Zusammenhang. Hinzu kommt das Problem der Quellenauswahl, das bei ständig steigender Zahl an Informationsquellen an Bedeutung zunimmt. Gerade wissenschaftliche Information verteilt sich auf die geschlossenen Angebote der Verlage und anderer Rechteinhaber, die ein wirtschaftliche Interesse an der Exklusivität des Zugangsweges haben.

Empirische Studien, wie die von O'Day u. Jeffries (1993) stützen das Modell des *Ber-*

rypickings, dass von Bates (1989) beschrieben wurde (s.o.)². Das Verhalten bei der Informationssuche besteht eher aus einer Serie von verbundenen, aber eigenständigen, Recherchen, die auf einem problemnahen Thema basieren. Neben der Recherche und der Iteration in Bezug auf ein Ziel tauchen nach O'Day und Jeffries zudem während der Informationssuche neue Ziele auf, so dass man zunehmend in völlig neue Richtungen navigiert und sucht. Das macht die Orientierung zusätzlich schwierig, so dass die Anwender vom Informationssystem dabei unterstützt werden sollten, ihre Strategie anzupassen und ihre verschiedenen Ziele im Blick zu behalten³. O'Day und Jeffries sprechen daher metaphorisch von Orientierungssport in einer Informationslandschaft – was die Sachlage noch immer gut beschreibt.

Informationssysteme müssen zudem beginnen, die Zusammenhänge zwischen den Informationen wieder herzustellen und so eine Orientierung und Navigation im Informationsraum zu ermöglichen. Die Zusammenhänge könnten beispielsweise auf Ähnlichkeitsberechnungen beruhen. Es könnten aber auch die speziellen Eigenschaften und Metadaten bestimmter Informationstypen genutzt werden. Wissenschaftliche Veröffentlichungen beispielsweise stehen in der Regel in festen Kontexten und Bezügen. Sie zitieren andere Veröffentlichungen oder werden zitiert. Die beteiligten Autoren kollaborieren gezielt, Ko-Autoren gelten daher als gute Quelle für Veröffentlichungen auf benachbarten Gebieten. Eine Navigation über solche impliziten Relationen wird *Multi Level Hypertext* genannt. (Fuhr 1990).

Um Multi-Level-Hypertext-Verknüpfungen zu berechnen und zu pflegen, ist ständige Arbeit des Informationssystems nötig. Neue Information muss in das bestehende Informationsnetz eingepflegt werden. Alte Verlinkungen müssen gepflegt und aktuell gehalten werden. Solche Updates können beispielsweise durch folgende Ereignisse ausgelöst werden:

- *Hat ein Autor eine neue oder mehrere Homepages?*
- *Haben Autoren zusammen veröffentlicht?*
- *Ist im Rahmen einer Konferenz oder eines Journals ein neues einschlägiges Dokument für einen relevanten Themenkomplex aufgetaucht?*

Ständige autonome Arbeit dieser Art ist gut geeignet als Einsatzfeld von Software-Agenten, ähnlich den *Spidern* oder *Bots*, die die Information für allgemeine Suchmaschinen beschaffen.

²Vergleiche auch Abschnitt 4.1.

³vergleiche (Hearst 1999, S. 264)

2.3 Verteilte Suche

Für die wissenschaftliche Arbeit ist es in der Regel wichtig, viele verschiedene digitale Bibliotheken für die Recherche zu benutzen. Verlage und Informationsanbieter pflegen meist ein abgegrenztes Informationsangebot. Integrationsbestrebungen in Form von virtuellen Fachbibliotheken wirken der Informationsverteilung zwar entgegen, aber immer neue Dienste und Angebote machen es schwierig, die verteilte Information über einen singulären Zugriffspunkt zu finden. Die verteilte Suche soll dafür sorgen, dass eine vom Benutzer formulierte Anfrage nicht durch die Verteilung und Anpassung an die Eigenschaften der unterschiedlichen Zugänge der Datenbanken ihre syntaktischen und semantischen Eigenschaften verliert. Dazu muss das Datenbank-, bzw. Metadatenschema der angebundenen Datenquelle mit dem generalisierten Schema der Meta-Suchsprache zur Deckung gebracht werden. Dies gilt insbesondere bei feldbasierter Metadatensuche, wie sie bei dem meisten digitalen Bibliotheken und OPACs⁴ gängig ist.

Während der verteilten Suche interessiert den Fragenden, wohin seine Anfrage verteilt wurde und wie viele Ergebnisse er jeweils zu erwarten hat. Die Ergebnisse möchte er einheitlich präsentiert sehen und möglichst nach einem einheitlichen Ranking sortiert. Beispielsweise muss entscheidbar sein, welche Quellen für die Recherche sinnvoll sind und herangezogen werden sollen. Wichtig ist, dass Metadaten über die Quellen verfügbar sind, oder aus den Inhalten berechnet werden können. Hierbei spielt es eine große Rolle, auf welche Weise die Inhalte der Datenquellen indexiert und zusammengefasst wurden (Nottelmann u. Fuhr 2001).

2.4 Heterogenität

Das Verteilen der Suchanfrage auf verschiedene Informationsangebote muss einen gemeinsamen Nenner bei der Metadaten-Erschließung der Datenquellen herstellen. Die bestehende Heterogenität der Angebote wirkt dem auf verschiedenen Ebenen entgegen. Die Anfrage sollte semantisch möglichst verlustfrei verteilt werden. Die Metadatenschemata müssen aufeinander abgebildet werden. Die semantischen Unterschiede der Suchterme je nach Datenquellen sollten behandelt werden. Manche

⁴ „Online Public Access Catalog“ - Öffentlich zugänglicher Online-Bibliothekskatalog

Datenquellen haben eine auf der Booleschen Logik basierende Anfragesprache, andere bieten einen probabilistischen Retrievalansatz.

Die Booleschen Anfragesprachen selektieren nach scharfer Ja/Nein-Unterscheidung nach Boolescher Logik, ob ein Dokument in die Ergebnismenge aufgenommen wird oder nicht. Eine konjunktive Boolesche Verknüpfung (AND) wird wie eine Schnittmengenbildung behandelt und eine disjunktive Verknüpfung (OR) zweier Teilanfragen wird als Vereinigungsmenge der Teilergebnisse zusammengeführt. Relevanzwerte für ein Ranking können bei diesem Ansatz nicht ohne weiteres ermittelt werden⁵.

Die probabilistischen Anfragesprachen ermitteln die Wahrscheinlichkeit, nach der ein Ergebnis relevant für die Suchanfrage ist. Da die zugrundeliegenden Algorithmen jedoch sehr unterschiedlich und in der Regel nicht bekannt sind, ist ein vereinheitlichter Ranking-Wert auch hier nicht über verschiedene heterogene Quellen berechenbar.

Neben dem uneinheitlichen Ranking ist auch die Metadaten-Heterogenität für den Anwender problematisch. Dem Benutzer ist die Heterogenität in Bezug auf sein Informationsbedürfnis zunächst nicht bewusst. Er möchte eine präzise und umfassende Antwort auf seine Anfrage und das möglichst schnell und effizient. Mechanismen, die die Komplexität der unterschiedlichen Metadaten mildern oder ganz verstecken, sind notwendig, um das zu erreichen.

Dies kann durch Standardisierung auf Seiten der Informationsanbieter geschehen. Es existieren verschiedene Standards für Metadaten wie z.B. das MARC Schema der amerikanischen Library of Congress, das vereinfachte "Dublin Core" Schema (DC), oder das Schema des Abfragestandards Z39.50 (Fox u. Sornil 1999).

Wenn die Heterogenität dennoch nicht vermeidbar ist, weil kein einheitlicher Metadaten-Standard existiert, dann kann das Problem durch spezielle Heterogenitätsbehandlung (siehe unten) gemindert werden. Selbst bei Schema-Ähnlichkeit kommt es in der Realität oft genug vor, dass verschiedene Varianten des selben Standards angewendet wurden, oder dass semantische Heterogenität bei der Verschlagwortung zu nicht optimalen Ergebnissen führt. Solche semantischen und pragmatischen Unterschiede können leicht dazu führen, dass die Integrationsleistung einer verteilten Suche stark verschlechtert wird.

⁵über zusätzliche Methoden, wie die Berechnung zentraler Autoren über einer Ergebnismenge oder das nachträgliche Berechnen von IDF-Werten über den Titeln der Ergebnismenge kann das ggf. nachgeholt werden (s.u.). (IDF steht hier für *Inverse Dokument Frequenz*: Ein im Index seltener Term führt zu höherer Gewichtung des Treffers)

Die Gegenmaßnahmen basieren meist auf statistischer oder probabilistischer Analyse, durch die das häufige gemeinsame Auftreten bestimmter Terme oder Phrasen analysiert wird, sowie der Nutzung von Thesauri und Crosskonkordanzen (Strötgen u. Kokkelink 2001; Krause 1999; Marx u. a. 1995). Crosskonkordanzen sind gerichtete, relevanzbewertete Relationen zwischen Termen zweier Thesauri oder Klassifikationen. Die verwendeten Relationsarten können hierbei Äquivalenzrelationen sein, Ober-/Unterbegriffsrelationen oder semantische Ähnlichkeitsrelationen. Crosskonkordanzen werden in der Regel intellektuell erstellt und sind daher aufwendig, dafür aber qualitativ hochwertig so dass sich dieser Ansatz vor allem für zentral gepflegte Fachdatenbanken anbietet (Walter u. a. 2006).

Der Anwender kann bei diesen Ansätzen entweder aus der Menge der errechneten Terme auswählen (Hellweg u. a. 2001), oder die Anfrage kann vom System automatisch ergänzt werden, um sie effektiver werden zu lassen.

2.5 Direkte Manipulation und Visualisierung in Informationssystemen

Die Recherchemöglichkeiten in den Informationssystemen der Informationsanbieter vor der Internetzeit, also bis in die 90er Jahre hinein, waren meist kommandosprachen- und textbildschirmorientiert. Suchen und Finden in solchen Datenbanken erforderte das Lernen und Beherrschen einer komplexen Anfragesprache und deren spezifischer Syntax. Bis heute sind solche Zugänge bei Hosting-Provider wie STN International oder GBI üblich, um in wohlerschlossenen Fachdatenbanken zu recherchieren.

Die folgende Beschreibung zu den beispielhaft genannten Hosts findet sich auf den Internetseiten der GESIS im Bereich des Informationszentrums Sozialwissenschaften, das seine Datenbanken SOLIS und FORIS über diese Hosts anbietet⁶:

STN International (The Scientific and Technical Information Network)

STN wird vom Fachinformationszentrum Karlsruhe, der American Chemical Society und dem Japan Information Center of Science and Technology als internationaler Rechnerverbund gemeinsam betrieben und ist einer der größten Datenbankanbieter für Wissenschaft und Technik. Alle STN-

⁶<http://www.gesis.org/iz/> (Stand Juli 2007)

Datenbanken sind über Datex-P oder via Internet in *STN on the Web* mit der Kommandosprache MESSENGER recherchierbar.

GBI (Gesellschaft für Betriebswirtschaftliche Information mbH)

Der Schwerpunkt im Angebot des GBI-Hosts liegt auf Datenbanken mit Informationen zu deutschen und internationalen Unternehmen, Produkten und Marktstudien, auf Pressedatenbanken sowie auf Datenbanken mit Informationen aus den Wirtschafts- und den Sozialwissenschaften. Die Datenbanken auf dem GBI-Host sind ebenfalls via Datex-P erreichbar und hier mit der Kommandosprache AOS abzufragen. Über den Web-Server der GBI sind alle Datenbanken auch über WWW-Suchmasken recherchierbar - wahlweise über die Schnell-Suche oder die Profi-Suche. Mit Wiso-Net bietet GBI die umfassendste Referenzsammlung zu den Wirtschafts- und Sozialwissenschaften im deutschsprachigen Raum, für die an nahezu allen Hochschulen ein Abonnement in der ein oder anderen Angebotsform besteht.

Die Recherchen waren und sind jeweils auch mit speziellen Kosten verbunden. Somit war diese Form der Recherche eher eine Experten-Angelegenheit und für den Durchschnittsnutzer kaum zugänglich.

Die Fortschritte von grafisch-direktmanipulativen Benutzerschnittstellen und der weiträumigen Vernetzung durch das Internet brachten also eine neue Zugänglichkeit der Information und den Recherchemöglichkeiten mit und beschleunigten den Aufbau der digitalen Bibliotheken. Auch die bereits diskutierte Informationsflut steht mit dieser Entwicklung in Zusammenhang. Dennoch brachte die technologische Entwicklung nur eine Teillösung mit sich, da die Recherchemöglichkeiten in ihrer Vielfalt und Komplexität wiederum die Nutzer überforderten oder wenigstens für die Mehrzahl der Anwender – für die der nun mögliche Zugang zur Information jetzt offen stand – sehr komplex blieb.

In diesem Abschnitt soll ein Überblick über die verwendeten Prinzipien und Methoden der direktmanipulativen Recherche/-oberflächen und Visualisierungswerkzeuge gegeben werden, um darzustellen, welche Möglichkeiten sich den Anwendern heute bieten.

Shneiderman (1997, S. 2) definiert eine direktmanipulative Benutzerschnittstelle über die folgenden Prinzipien:

- Kontinuierliche Präsentation der Objekte und Aktionen, die für den Benutzer von Belang sind.
- Statt komplexer Syntax werden alle Aktionen durch direkte Manipulation von Objekten oder durch Schaltflächen mit Beschriftung angeboten.
- Es gibt unmittelbare Operationen auf Objekte, deren Wirkung sofort sichtbar wird und die wieder rückgängig gemacht werden kann.

Diese relativ breit gefasste Definition deckt also bereits Systeme ab, die auf Menü-Strukturen und Formularen als Eingabemöglichkeit beruhen – direktmanipulative Benutzungsoberflächen können aber ebenso Gebrauch von komplexen Visualisierungen sowie Ikonen und Metaphern machen, um den Anwendern die direkte Manipulation von Objekten oder Graphen anzubieten. Da der Gestaltungsspielraum sehr groß ist, lohnt es sich, die möglichen Interaktionsmechanismen zu differenzieren und auf ihre Tauglichkeit zu prüfen, insbesondere in Hinblick auf die Gestaltung von Recherche- und Informationssystemen.

2.5.1 Formulare

Die Formularmetapher wurde auch auf den textbasierten Terminals (TTY) der siebziger und achtziger Jahre schon erfolgreich in Form von Eingabemasken neben den Kommandosprachen eingesetzt.

Ein gut gestaltetes Formular verspricht folgende Vorteile: Die Anwender bekommen vorgelegt, für welche Felder Eingaben möglich sind. Zudem können die Eingabefelder je nach möglichem Datentyp passende Eingabeelemente (sogenannte Widgets) verwenden, um zum Beispiel die Eingabe einer Zahl oder eines Datums komfortabel zu gestalten. Felder können mit einer Vorlagefunktion versehen werden. Benutzereingaben werden validiert und auf Konsistenz geprüft. So werden Eingabefehler verhindert. Formulare sind aus der Alltagswelt bekannt und daher sind sie in hohem Maße selbsterklärend. Die Verhaltensweisen bei den vorangegangenen Mehrwertfunktionen sind jedoch schon eine Besonderheit der Computer-Umsetzung und nicht mehr selbsterklärend. Daher ist eine Einhaltung relevanter Styleguides zur Erreichung hoher systemweiter Konsistenz unumgänglich.

Die Formulare eignen sich in Hinblick auf die Recherche gut, um Anfragen zu spezifizieren. Die Felder bilden sich dann auf Anfragebedingungen ab, meist als Feldname, Operator und Wert. Die Felder werden bei der Abbildung auf logische Aus-

drücke implizit durch logische Konjunktion miteinander verknüpft. Wenn mehrere Terme in den Wert-Eingabe-Bereich des Formulars eingegeben werden können, entspricht dies meist einer disjunktiven Logik (ODER). Wenn komplexere Boolesche Logik abgebildet werden muss, wird die reine Formularmetapher meist nicht ausreichen, insbesondere in Hinblick auf die Flexibilität, die bei komplexen geklammerten Ausdrücken von Nöten wäre.

Dynamische Formulare, die Struktur und Aufbau, sowie Beschriftung der jeweiligen Booleschen Struktur der spezifizierten Anfrage anpassen sind zwar möglich, jedoch viel komplexer und metaphorisch nicht mehr selbsterklärend. Die eingeschränkte Flexibilität verbietet, diesen Modus als alleinige Gestaltungsform zu verwenden. Die vielfältigen Vorteile machen Formulare jedoch weiterhin attraktiv, allerdings nicht als alleiniges Gestaltungsmittel.

Zur Diskussion der Vorteile einer formularbasierten Oberflächengestaltung siehe auch die Diskussion zum Formularmodus des WOB-Modells in Krause (1997a, b) sowie die Abschnitte 4.3 und 4.5.3 zur Umsetzung in DAFFODIL.

2.5.2 Menüsteuerung

Menüsteuerung ist ein Bedienelement, dass seinen Weg in die meisten direktmanipulativen Oberflächen gefunden hat. Seine Verwendung resultiert aus der Beobachtung, dass externalisiertes Wissen und Wiedererkennung die kognitive Last im Vergleich zur Kommandosprache reduziert („Vorlageleistung“).

In empirischen Untersuchungen (Krause 1997b, siehe Abschnitt "Menüs", S. 14) lässt sich jedoch nachweisen, dass Menüsysteme schnell an die Grenzen menschlicher Kognition stoßen:

Auch nach längerer Eingewöhnungszeit gelingt es Benutzern nicht problemlos, Handlungsoptionen im Menü aufzufinden, richtig anzuwenden und die darin enthaltenen Elemente der Zustandsanzeige adäquat zu interpretieren. Unter anderem sind hierfür die Benennung der Menüeinträge, die Gruppenbildung und die Zustandsanzeige durch den Häkchenmechanismus verantwortlich.

Empirisch nachweisbar sind vielfältige Probleme der Benennung der Menüpunkte, die zu vielen Missverständnissen, sowie zu Lokalisierungsproblemen führen können. Auch Verständnisprobleme mit den unterschiedlichen Typen von Menüeinträgen,

die konventionell vorhanden sind (Häckchenmetapher, mit Ikonen versehene Menüpunkte, Menüpunkte mit Ellipse zum Aufruf von Dialogen...) lassen sich kaum vermeiden.

Die vielfältigen Nachteile führen zu einer generellen Ablehnung der Menüsteuerung im Rahmen des WOB-Modells (siehe Abschnitt 4.3) und werden daher im Rahmen dieses Gestaltungsmodells nicht für die ergonomische Gestaltung von Informationssystemen empfohlen.

Wo es sinnvoll ist, können gegebenenfalls Kontextmenüs eingesetzt werden und wenn der Style-Guide der Systemumgebung dies verlangt, sollten die Mindestanforderungen evtl. erfüllt werden, um Konsistenz mit anderen Applikationen zu wahren.

2.5.3 Ikonen, Maus und Zeiger

Die Repräsentation von manipulierbaren Objekten als Ikonen und die Verwendung einer Maus zur Steuerung eines Zeigers dominiert heutige Betriebssystem-Oberflächen. Zusammen mit den Menüs ergibt sich das sogenannte *Windows, Icons, Menus and Pointers* (WIMP) Schema.

Die ikonische Darstellung beruht meist auf einem metaphorisch zugeordneten Bild, das mit dem Mauszeiger aktiviert und manipuliert werden kann. Ein Dokument wird etwa als ein Blatt Papier dargestellt, ein Ablageordner als Mappe usw. Die Metaphorik orientiert sich bei der Schreibtischumgebung von Betriebssystemen an der Büro-Alltagswelt. Es gibt auch sogenannte Funktionsobjekte, wie Drucker oder Papierkörbe, die eine Funktion ausführen, wenn ein Datenobjekt mit der Maus darübergezogen und fallengelassen wird.

Solange die Metaphorik trägt, ist die Selbsterklärungsfähigkeit bei dem WIMP-Schema sehr hoch und der Anwender kann sein Alltagswissen gut transferieren. Bei vielen abstrakteren Funktionen wird es jedoch schwierig, tragfähige Metaphern aus der Alltagswelt heranzuziehen. Wenn die Metaphern brechen, ist es oft für den Anwender schwer, den Lernschritt nachzuvollziehen.

Es kann auch vorkommen, dass ein Einhalten der strikten Metaphorik zu einer großen Flut von Funktionsobjekten führt, die wiederum auf Kosten der Erlernbarkeit und Benutzbarkeit geht. Die Schreibtischoberfläche N/JOY beispielsweise hat eine strikte Interpretation der objektorientierten Manipulation unter Verwendung von Funktionsobjekten implementiert. Statt etwa einen Text-Stil aus Parametern in Vorlagelisten zu-

sammenzustellen und auf eine Text-Selektion anzuwenden, wurden die Stilelemente als Stifte dargestellt, die parametrisiert und auf Textstellen angewendet werden konnten. Bei Verwendung vieler Stilvarianten konnte es zu einer unübersichtlichen Menge an Stiften kommen, deren Benutzbarkeit hinter die herkömmliche Variante mit Menüs und Auswahllisten zurückfiel.

In Informationssystemen ist es oft nicht einfach, die Elemente zu definieren, die sich als Daten- oder Funktionsobjekte eignen und zu tragfähigen Metaphern führen („Was ist ein Dokument?“). So ist es noch gut möglich, den Autoren eines Textes als *Personenobjekt* ikonisch zu repräsentieren, bei URLs oder dem Veröffentlichungsjahr und anderen Metadaten ist das jedoch nicht gerechtfertigt. Die Gestaltung orientiert sich meist an der objektorientierten Modellierung, wobei es nicht einfach ist, die Benutzersicht vor zu vielen technisch motivierten Details zu bewahren – im Sinne einer nutzerorientierten Herangehensweise. Die richtige Konzeption ist wichtig, damit nicht unnötig Bedienhemmnisse aufgebaut werden.

Zur Diskussion der Objektorientierung in Hinblick auf ikonisch repräsentierte Objekte vergleiche (Krause 1997b, Abschnitt 4.2, S. 29 ff.)

2.5.4 Visuelle Formalismen

Neben den metaphorischen Ikonen sind andere grafische Elemente wichtige Gestaltungsmittel direktmanipulativer Oberflächen. Solche Elemente fallen häufig in die Klasse der visuellen Formalismen (*Visual Formalisms*), die bei Krause (1997b, S. 20) wie folgt definiert sind:

Visual Formalisms sind im Kern nichtbildhafte, nichtmetaphorische, visuelle Gestaltungsmittel wie Tabellen, Spread Sheets, Liniengraphiken oder hierarchische Baumstrukturen, deren grafischer Charakter in Verbindung mit den kognitiven Grundfähigkeiten des Menschen (Raumwahrnehmung u.a.) eine effiziente, direktmanipulative Systembedienung ohne (bzw. mit nur geringem) Lernaufwand ermöglichen und die Problemlösung durch „external representation“ unterstützen.

Diese Definition geht zurück auf eine Charakterisierung von Nardi u. Zarter (1993), die auf der Kritik an der bis dahin vorherrschenden Metaphernbildung basierte. Das WOB-Modell verbindet beide Konzepte und adressiert so die widersprüchlichen An-

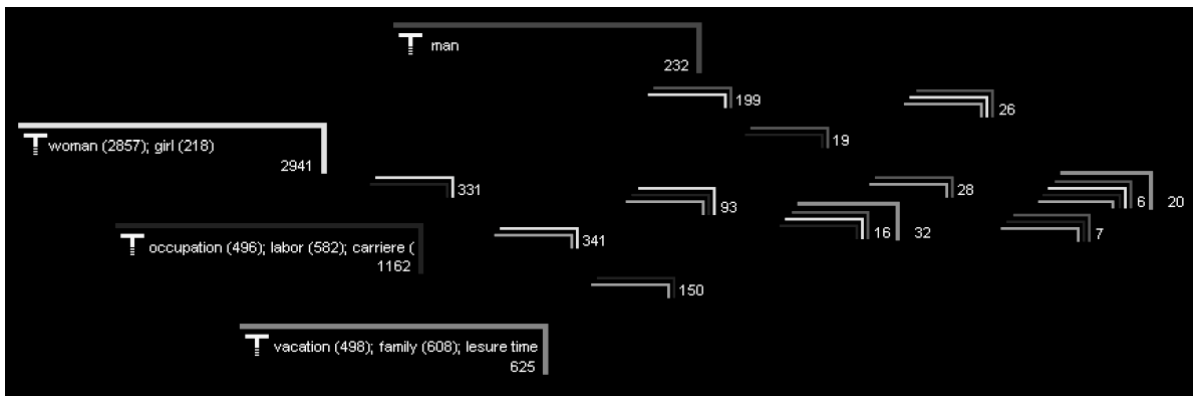


Abbildung 2.3: DEViD: Visueller Formalismus zur Repräsentation der Ergebnismengen bei Boolescher Anfragekombinatorik.

forderungen an sehr weitgehende Flexibilität der Gestaltung und klare Strukturierung der Dialoge, sowie an klare Dialogleitlinien.

In Stempfhuber (2001) werden visuelle Formalismen angewendet, um strukturelle und semantische Heterogenität bei der verteilten Recherche in Text- und Faktendatenbanken in Hinblick auf Anfragespezifikation und den Navigieren in den Ergebnissen ergonomisch zu gestalten.

In Eibl (2000) wird mit DEViD ein effizienter und auch ästhetisch ansprechender Recherchemodus auf Basis visueller Formalismus entwickelt, der die Anwender bei der Verwendung Boolescher Recherchelogik entlastet, siehe Abbildung 2.3. Der Anwender gibt in die durch farbige Klammern markierte Eingabefelder Suchworte ein und das System berechnet für die boolesche Kombinatorik dieser Terme die jeweiligen Treffermengen – die durch die farbigen Klammern einen Rückbezug auf die Eingabefelder herstellen. Der Anwender kann die Ergebnismengen anschließend explorativ inspizieren.

2.5.5 Graphen

Ein häufiger Sonderfall visueller Formalismen sind Graphen, die Datenreihen oder Zusammenhänge zwischen einzelnen Daten visualisieren. Dies kann beispielsweise in Form von Liniengraphiken geschehen, oder durch mit Kanten verbundenen Knoten in einer Graphen-Struktur. Viele weitere grafische Darstellungen von Daten sind denkbar und werden auch verwendet. Einige Beispiele sind im Folgenden kurz dar-

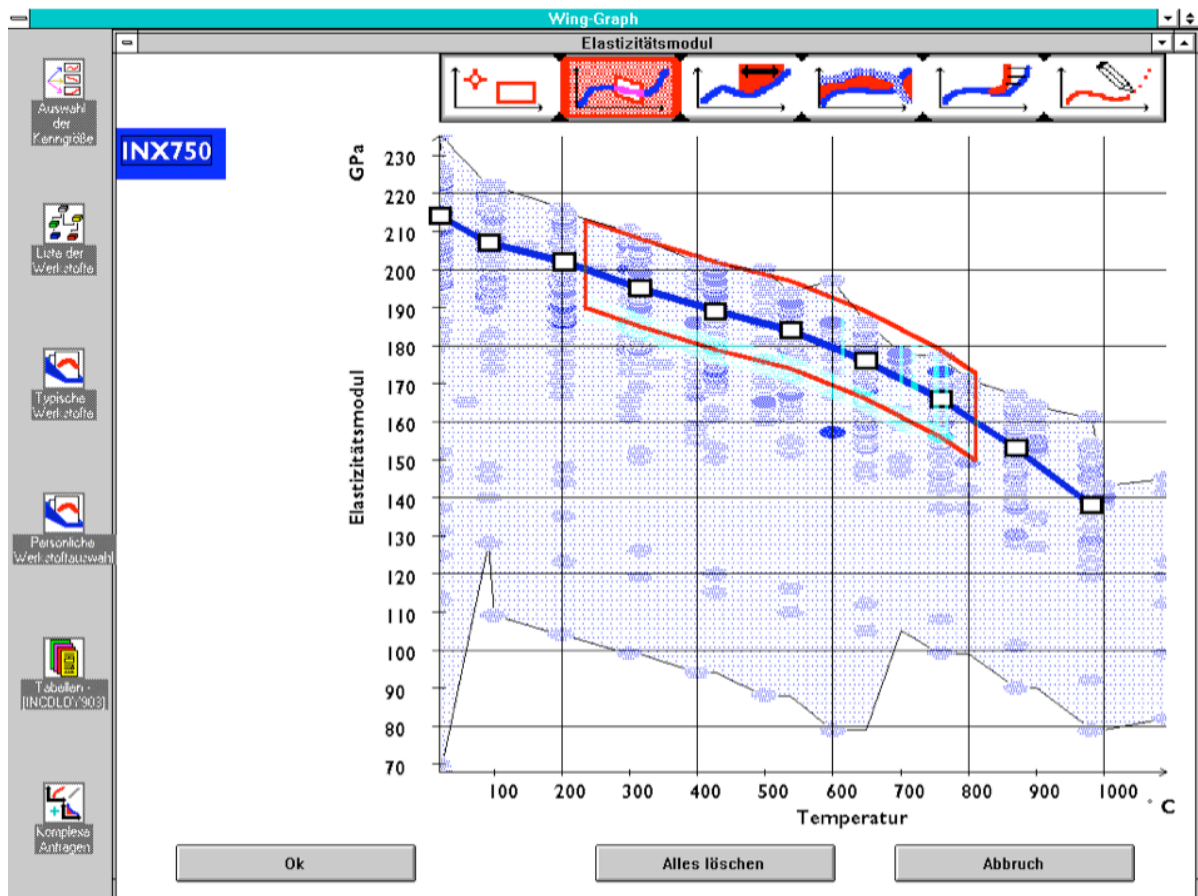


Abbildung 2.4: Elastizitätsmodul in WING-M2

gestellt.

Interessant ist die Erkenntnis im Rahmen der Evaluationen im Projekt WING an der Uni Regensburg, dass die Nutzer den Wechsel der Repräsentation in einen derartigen grafischen Modus (siehe Abbildung 2.4 aus Wolff (1996)) gut nachvollziehen können, jedoch fortan in Parametern des Graphen weiterdenken und eine Rückkehr zu einer textuellen Darstellung als Hindernis empfinden. Es ist ein großer Vorteil, wenn die Anwender bei der Recherche den Graphen direkt manipulieren können, um die Parameter in die gewünschte Rechercherichtung zu ändern.

Von Roppel (1995) wird ein Konzept entwickelt, um auch komplexe Baumstrukturen unter Berücksichtigung des beschränkten Bildschirmplatzes effizient navigieren zu können, indem der Baum kontextbezogen beschnitten wird und entsprechende Navigationselemente zum Absteigen und Umsteigen in Seitenarme angeboten werden (ohne Abbildung).

In Mutschke (2001) wird ein Konzept vorgestellt, um Autorennetzwerke, gebildet aus Ko-Zitationsbeziehungen, explorativ zu erschließen und die jeweils nächstliegenden zentralen Personen visuell erfassbar zu machen. Siehe Abbildung 2.5 und zur Umsetzung in DAFFODIL siehe Abschnitt 4.18.

Zusammenfassend lässt sich sagen, dass die verschiedenen Methoden und Ansätze, die sich zur Erstellung einer direktmanipulativen Benutzerschnittstelle anwenden lassen, am besten in einer multimodalen Kombination wirken und sich gegenseitig gut ergänzen.

2.5.6 Visualisierung

Die visuellen Formalismen, insbesondere in Form von Graphen werden oft auch zur Visualisierung von inhärent abstrakten Datenmengen eingesetzt. Besonders in Bezug auf textuelle Darstellung stellt die Visualisierung als Zugangsweg zu Fundstellen im Recherchekontext ein besonders schwieriges Feld dar.

Hearst (1999) nennt folgende Prinzipien, die in Systemen, die mit Visualisierung arbeiten zum Einsatz kommen:

- *Brushing and Linking*

Mehrere Ansichten (Views) auf die selben Daten werden parallel angeboten. Manipulation in einer Ansicht betrifft auch die Darstellung in den weiteren Ansichten. So können auch textuelle Listendarstellungen und Visualisierungen gekoppelt werden. Eine Anwendung dieses Prinzips findet sich beim Projekt VisMeB (Reiterer u. a. 2003), siehe auch Abbildung 2.6.

- *Panning and Zooming*

Unter *Panning and Zooming* versteht man eine Interaktionsart die sich wie eine Filmkamera verhält, die immer nur Ausschnitte aus der Gesamtdatenmenge zeigt, aber sich über das Datenfeld bewegen kann und durch Heraus- und Hereinzoomen in die Daten sowohl Überblick als auch Detailtreue herstellen kann. Eine populäre Implementation stellt Googles Webanwendung *Google Maps* (siehe Abbildung 2.7) sowie das drei dimensionale geographische Informationssystem *Google Earth*, zum Suchen und Finden geografischer Information, dar.

- *Focus-Plus-Context*

Bei *Focus-Plus-Context* wird eine verzerrte Darstellung gewählt, bei der die der-

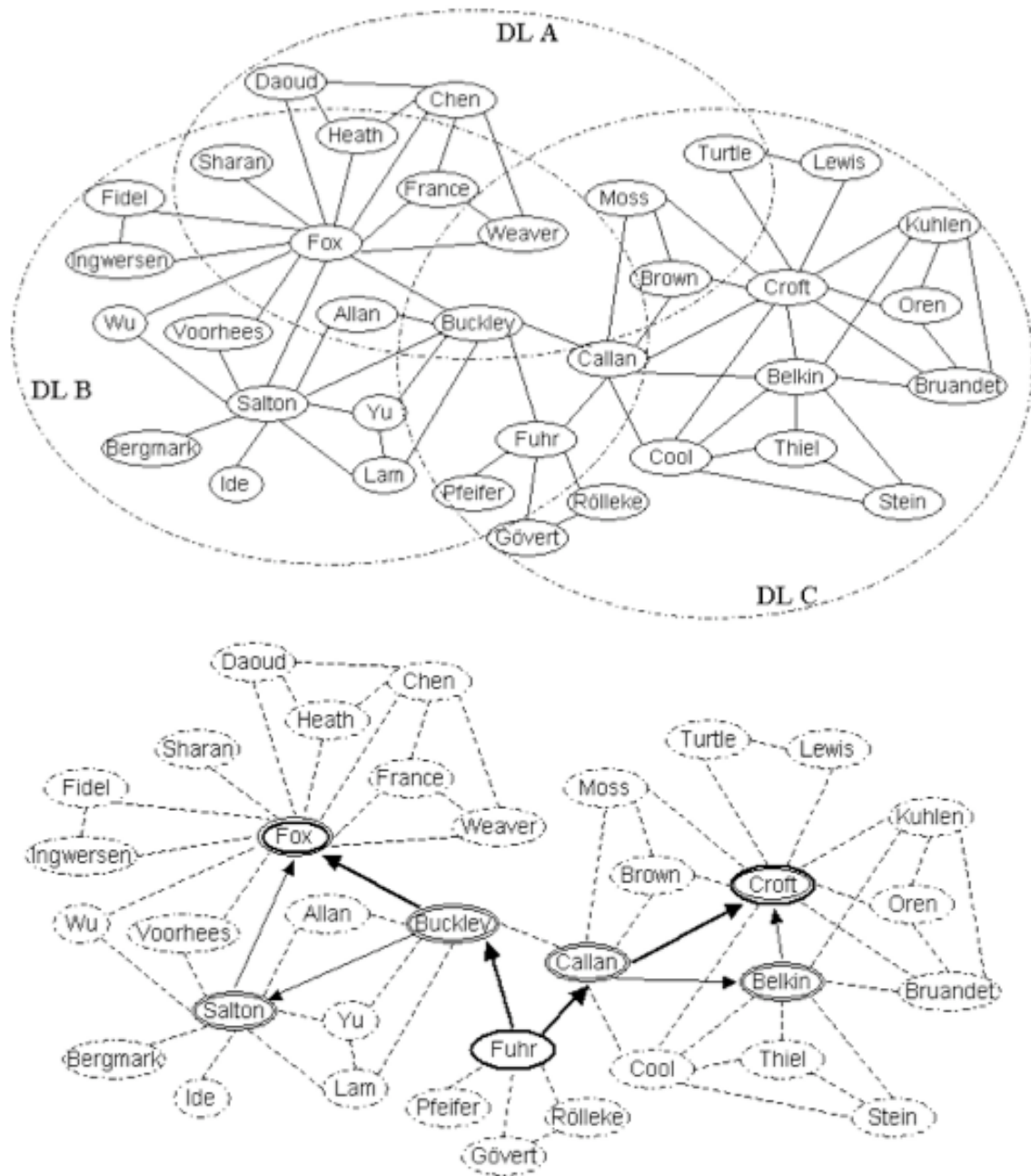


Abbildung 2.5: Autorennetzwerk, berechnet über drei digitale Bibliotheken, und Hub-Berechnung, nach Mutschke (2001).

2 Information Retrieval in verteilten digitalen Bibliotheken

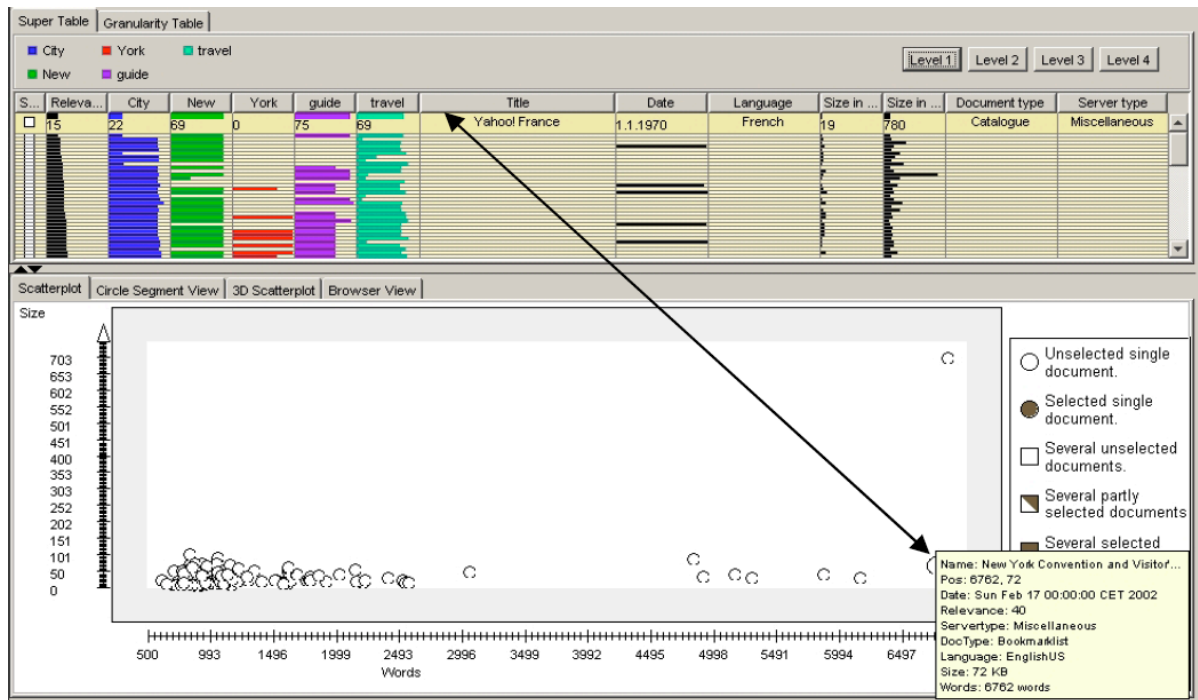


Abbildung 2.6: Brushing and Linking bei VisMeB

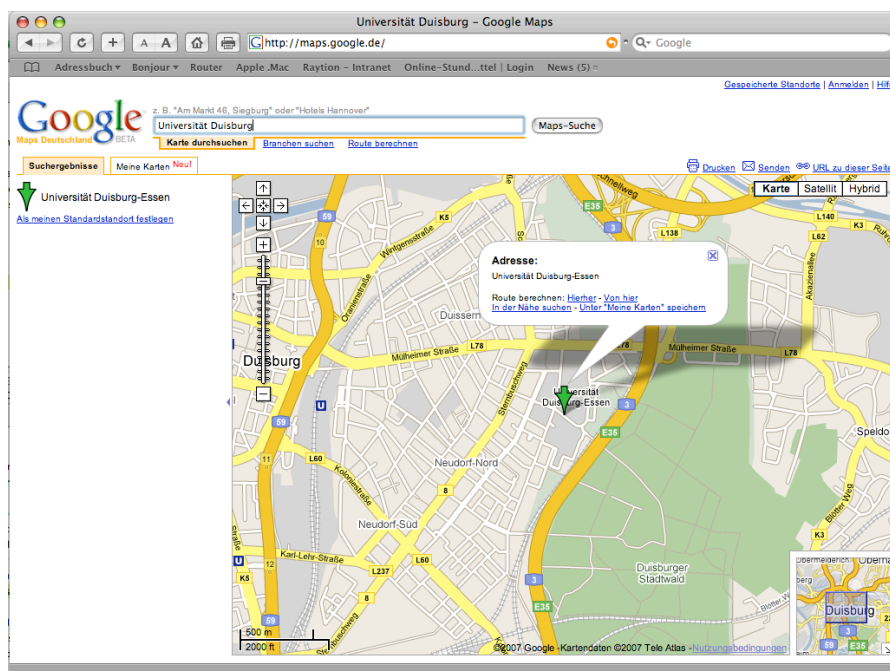


Abbildung 2.7: Die Google Maps Applikation

zeit im Focus befindliche Information herausgehoben (z.B. vergrößert) dargestellt wird und die umliegenden, in Bezug stehenden Daten, werden so skaliert, dass sie zwar navigierbar bleiben, aber in den Hintergrund treten. Wichtig ist, dass neben dem Fokus auch der Kontext erhalten und sichtbar bleibt. Einen Überblick über entsprechende Visualisierungen bieten (Leung u. Aerley 1994).

- *Magic Lenses*

Magic Lenses sind direkt manipulierbare Fenster, die sich wie Linsen über eine Visualisierung bewegen lassen und dabei eine Darstellungstransformation durchführen.

- *Animation*

Animation wird in zunehmendem Maße eingesetzt, um zum Beispiel bei Panning-And-Zooming Mechanismen den Kontext für den Benutzer deutlich ineinander übergehen zu lassen und so zu helfen, die Orientierung zu behalten. Auch dynamisch verdeckte Information kann animiert beiseite geschoben werden, um Hinweise zu geben, wie der Anwender wieder an die Information, die nun verdeckt ist, kommen kann.

Ein weiterführender und tiefer gehender Überblick über Visualisierungsbeispiele und Ansätze finden sich bei Eibl (2000) und Hearst (1999). In Bezug auf Visualisierungsansätze für das Textretrieval bietet auch (Mann 2002) einen guten Überblick.

2.5.7 Spezifika für das Textretrieval

Wie Hearst (1999) richtig feststellt, stellt das Textretrieval, das für den Bereich der digitalen Bibliotheken vorrangige Bedeutung hat (wenn auch Medien und Faktenretrieval zunehmend an Bedeutung zunimmt, da die notwendige Digitalisierung schnell voranschreitet), besondere Anforderungen an die Retrievalschnittstelle, denn Daten sind schwierig zu visualisieren, oder als Objekte zu repräsentieren.

Der meistvertretene Ansatz zur Realisierung von Rechercheoberflächen für das Textretrieval bleibt nach wie vor die Kombination eines Formulars zur Anfragespezifikation und eine listenförmige Anzeige der Ergebnisse mit Mechanismen zur besseren Handhabung der Ergebnisliste. Die Empfehlungen von Byrd u. a. (1998, S. 6) bleiben somit gültig, eine Schnittstelle für das Textretrieval sollte mindestens folgende Aspekte berücksichtigen (siehe auch Abbildung 2.8):

1. Formulation:

2 Information Retrieval in verteilten digitalen Bibliotheken

- Sources: specify which libraries and/or collections to search and the search range within them.
 - Fields: specify which text fields are to be searched. Searches may also be restricted by structured fields.
 - What to search for: text selected or typed by user, perhaps as one or more phrases.
 - Variants: control over features like case sensitivity, word stemming, partial matches, phonetic variants, stop words, synonyms, abbreviations, and broader or narrower terms. In all cases, the user interface should make it clear which variants, if any, are allowed.
2. Action: how does a search get initiated explicitly (e.g., with a button), or implicitly (e.g., when some aspect of the query is changed).
 3. Review of results: conventional options are, for example, to specify result set size, layout, sequencing (alphabetically, chronologically, relevance ranked, etc.), and contents (which parts and fields are displayed). Less conventional interfaces might employ a wide variety of techniques, including many based on information-visualization research.
 4. Refinement: provide feedback on search results with informative messages and clustering of results. For example, enable progressive querying, especially with relevance feedback; history keeping; and extraction of results to files, perhaps for use in e-mail.

Netscape: NewsFinder

NewsFinder

Sources
(Number of articles available)

Select the newspapers you wish to search:
☐ New York Times (158,301)
☐ USA Today (72,323)
☐ Washington Post (91,372)
☐ Wall Street Journal (93,234)
☒ **All Newspapers**

Set the date range for search:
 Year 92 93 94 95 96 97 98

☐ **All Dates**

Select the sections you wish to search:
☐ National ☐ International ☒ Arts ☐ Business ☐ Sports
☒ **All Sections**

Fields
Select the portion(s) of the article you would like to search:
☐ Headline ☐ Byline ☐ First two paragraphs ☐ Captions
☒ **Entire Article**

Variants
☐ Synonyms ☐ Words with same stem ☐ Abbreviations ☒ Ignore case

Phrases
(For example: Bill Clinton, air pollution control)
 modern art
 museum exhibits

Results
Order by: ☐ Newspaper ☐ Date ☐ Section ☒ Relevance
Include: ☒ Headline ☐ Byline ☒ Newspaper ☐ Section ☒ Date

Action

Abbildung 2.8: Prototypisches Textretrieval-Interface nach Byrd u. a. (1998).

2.6 Agenten

Bei der Anwendung von Agententechnologie im Kontext von verteilten digitalen Bibliotheken stellen sich eine Reihe spezifischer Probleme, die in einfacheren Anwendungsdomänen so nicht vorkommen. In den folgenden Abschnitten werden einige Maßnahmen genannt, die diese Probleme adressieren.

2.6.1 Mehrwertdienste

Die Recherche nach wissenschaftlicher Information ist eine wissensintensive Tätigkeit, mit vielen komplexen semantischen Zusammenhängen.

Inhaltliche Relationen wie etwa bibliometrische Zitationsbeziehungen oder soziale Relationen können aus den Daten bestimmt werden, um so informationellen Mehrwert zu erhalten.

Anwender nutzen im Idealfall diese Zusammenhänge, um strategisch nach der gesuchten Information zu recherchieren. Experten und erfahrene Benutzer kennen den Wert, der sich durch solche Mehrwertdienste ergibt und nutzen diese für ihre Recherchen. Agenten können im Hintergrund bei Aufbau und Pflege der Wissensbasen wirken und dann die Mehrwertdienste und hochwertige Suchfunktionen anbieten.

Der Benutzer hat so den taktischen Vorteil, mit relativ einfachen Anfragen beginnen zu können und im Zuge der Iteration und Anfragerreformulierung die angebotenen Navigationshilfen nutzen zu können, um schneller und gezielter zu den eigentlich einschlägigen Informationen zu gelangen. Bei jedem Schritt können die Mehrwertdienste gezielt erfolgversprechende Rechschritte, sinnvolle Filter und Anfragetermine anbieten und den Suchenden so leiten.

2.6.2 Strategische Unterstützung

Bei Laien ist die zum effektiven Nutzen dieser Dienste notwendige Erfahrung oder der nötige Überblick über das Themengebiet in der Regel nicht vorhanden, sie brauchen daher Anleitung und Unterstützung bei taktischer oder strategischer Herangehensweise an die Recherche.

Proaktive, agentengestützte Funktionen können sowohl bei der Anfrageformulierung als auch bei der Verknüpfung der über verschiedene Dienste verteilten Information

helfen. Auch häufig vorkommende Taktiken, die sich als automatisierbar herausstellen, können so als Baustein der Recherchestrategie angeboten werden – etwa die Suche nach einem Volltext, für den nur Fragmente der Metadaten bekannt sind (*Known Item Instantiation*).

2.6.3 Monitoring und Awareness

Die Beobachtung eines Themas (*Monitoring*) über stehende Anfragen und die Benachrichtigung oder Markierung neuer Objekte durch Awarenessfunktionen ist eine wichtige Unterstützung für Anwender, die gut mit Software-Agenten realisiert werden kann.

Benutzer organisieren ihre Sammlung von Metadaten und möchten evtl. über dieses Thema auch in der Folgezeit auf dem aktuellen Stand gehalten werden.

Agenten können die einschlägigen Quellen (Journale, Konferenzen, Web-Portale) beobachten und potentiell Relevantes zusammentragen. In kollaborativen Informationssystemen können auch weitere Benutzer neue relevante Objekte in einen gemeinsamen Speicherbereich legen. Auf solche Änderungen möchten Benutzer aufmerksam gemacht werden.

2.6.4 Integration, Homogenisierung und Filtern

Die Anbindung von verschiedenen Diensten, Wissensquellen und digitalen Bibliotheken an ein lose gekoppeltes Informationssystem, das einen konsistenten und integrierten Zugang zu wissenschaftlicher Information bietet, lässt sich mit verteilten Software-Agenten, so genannten *Information Agents* und *Wrappern* bewerkstelligen.

Diese vermittelnden Agenten können die verfügbare Information homogenisieren, etwa in ein gemeinsames Datenschema überführen. Auch Retrievalfunktionen oder Filter die von den angesprochenen Hintergrund-Diensten so nicht angeboten werden, können von den vermittelnden Software-Agenten und Diensten als Mehrwert hinzugefügt werden.

2.7 Gemeinsame Anwendung

2 Information Retrieval in verteilten digitalen Bibliotheken

Die oben beschriebenen Hauptaspekte des Information Retrieval in verteilten digitalen Bibliotheken sind bereits anderweitig beschrieben und werden eingesetzt. Es ist jedoch bisher kein Informationssystem zur Recherche in digitalen Bibliotheken bekannt, das all diese Methoden und Techniken gemeinsam in ihrer Gesamtheit verwendet und zum Nutzen des Anwenders verfügbar macht.

Wie diese Funktionen im Einklang mit dem softwareergonomischen WOB-Modell (siehe Abschnitt 4.3) gebracht werden können, ist ebenso Gegenstand dieser Arbeit. Das WOB-Modell geht von einer objektorientierten Werkzeugmetapher aus. Es trägt dazu bei, dass softwareergonomische Grundprinzipien bestmöglich innerhalb einer Benutzungsoberfläche eines integrierten Informationssystems realisiert werden können.

3 Arbeitsteilung in agentengestützten Informationssystemen

Im folgenden Kapitel wird ein Überblick über den Stand der Technik auf den relevanten Gebieten *Agenten*, *Interaktionsparadigmen*, *Oberflächengestaltung für Informationssysteme* sowie *Mentale und Kognitive Modelle* gegeben.

3.1 Proaktivität versus direkte Manipulation

Seit etwa zehn Jahren wird über einen Wandel der vorherrschenden Interaktionsparadigmen diskutiert. Offensichtlich wird das bisher dominierende Paradigma der *direkten Manipulation* allmählich abgelöst, oder zumindest in Teilbereichen verdrängt.

Bei der direkten Manipulation geht die Initiative vom Benutzer aus. Dieser interagiert mit den Objekten und Kontrollelementen einer grafischen Oberfläche, die meist dem *Windows, Icons, Mouse and Pointer* Konzept entspricht. Der Benutzer erlebt so alle Funktionen des Systems als direkte Folge seines Handelns. Wendet er Funktionen auf Objekte an, selektiert er diese mit der Maus und wählt die Funktion durch ein Menü (schwache Objektorientierung, vgl. Krause (1997b)) oder durch Ziehen und Fallenlassen der Objekte auf Funktionsobjekte (starke Objektorientierung, ebenda). Die Softwaresysteme verhalten sich rein passiv, bis eine Aktion vom Benutzer angestoßen wird.

Neuere Softwaresysteme verhalten sich hingegen zunehmend *proaktiv*. Das heißt sie übernehmen einen Teil der Initiative und stoßen ihrerseits automatisch Prozesse an. Insbesondere das Konzept von *Software-Agenten* hat eine Reihe von Diskussionen und Entwicklungen in diese Richtung gehend ausgelöst.

So entstehen Softwaresysteme mit gemischter Initiative (*Mixed Initiative User Interfaces, MIUI*) (Horvitz u. a. 1999). Da die Anwender in diesen Systemen einen Teil der Kontrolle an das System abgeben, muss abgewogen werden, wie viel Kontrolle das System übernehmen kann und soll. Die Autonomie des Benutzers hat einen engen Zusammenhang mit seiner Zufriedenheit und Effektivität (siehe auch Abschnitt 3.2.3). Bei steigender Komplexität der Anwendung kann es jedoch notwendig sein, proaktives Verhalten in der Gestaltung zu berücksichtigen, um überhaupt ein ergonomisch sinnvolles System bauen zu können. In diesem Spannungsfeld entstehen immer wieder Zielkonflikte.

Zielkonflikt heißt im Sinne der Softwareergonomie, dass sich zwei Gestaltungsziele gegensätzlich gegenüberstehen und sich eines davon in der Regel nur durch Abstriche auf der anderen Seite erreichen lässt. Meist muss deshalb ein sinnvoller Kompromiss gefunden werden. Ein klassischer Zielkonflikt tritt bei der Gestaltung von Tastaturen auf: Auf Tastaturen mit enger stehenden Tasten kann schneller getippt werden. Gleichzeitig steigt jedoch die Tippfehler-Quote. Will man also eine Tastatur entwerfen, auf der man gleichzeitig schnell und fehlerfrei tippen kann, muss man die dafür optimale Tastengröße ermitteln. Die Gestaltungsform der gebräuchlichen Tasten einer Tastatur verbessert beides: Durch die nach oben schmaler werdenden Tasten, die leicht konvex sind, können die Tasten enger stehen, ohne die Fehlerhäufigkeit zu groß werden zu lassen.

Den Gedanken des Zielkonfliktes hat auch die Softwaretechnik genutzt, um häufig vorkommende *Entwurfsmuster* in der objektorientierten Programmierung zu beschreiben (Gamma u. a. 1995). Dort werden die gegensätzlich wirkenden Ziele metaphorisch als „Kräfte“ bezeichnet, die ausgeglichen werden müssen. Das Entwurfsmuster hat demnach die Eigenschaft, den Zielkonflikt zu lösen. Es kann mehrere Varianten geben, bei denen der Schwerpunkt im Kräfteverhältnis verlagert wird, und somit andere Design-Entscheidungen bedingt.

3.1.1 Direkte Manipulation versus Software-Agenten

Die Befürworter des Paradigmas „direkte Manipulation“ sehen das Konzept von proaktiver Software oder Software-Agenten kritisch, da es dem Grundsatz zuwider läuft, nach dem alle Initiative der Interaktion vom Benutzer ausgeht. Ben Shneiderman hat in einer Debatte mit Pattie Maes, als Vertreterin des Agenten-Paradigmas, diesen Standpunkt beschrieben (Shneiderman u. Maes 1997):

Das Ziel des User-Interface-Designs sei es, Umgebungen zu schaffen, in denen Anwender die Anzeige verstehen, sich das System auf vorhersagbare Weise verhält und Anwender die Verantwortung für Aktionen übernehmen. Die Kontrolle, die Initiative und die Verantwortlichkeit für Handlungen sind nach Shneiderman zentrale Begriffe für gutes Anwendungsdesign. Eine gute Analyse der Hauptaufgaben der Anwender ist essentiell, um ihnen das richtige Maß an Kontrolle und die richtigen Kontrollelemente für die entscheidenden Parameter an die Hand zu geben. Für komplexe Domänen sei die Informationsvisualisierung sehr wichtig, um Anwendern die Grundlage für richtige Entscheidungen an die Hand zu geben, zum Beispiel den richtigen Überblick über große Datenmengen.

Maes hingegen charakterisiert die Entwicklung von Software-Agenten als neuen Ansatz für Anwendersoftware. Anstatt Software nur passiv, ohne eigene Initiative zu gestalten, wird es mit Software-Agenten konzeptuell möglich, proaktive Software in eigenständigen, personalisierten Teilsystemen zu entwickeln, die Anwender bei ihren Aufgaben helfen, indem sie deren Interessen kennenlernen und beachten. Zudem können Agenten langlebiger gestaltet werden, so dass sie mit der Zeit *lernen* können (Adaptivität) und auch autonom handeln können, ohne im direkten Kontakt mit den Anwendern zu stehen. Sie begründet die Notwendigkeit, Agenten und proaktive Softwarekomponenten zu benutzen, mit der steigenden Komplexität der Arbeitsumgebungen (Maes 1994) bei gleichzeitig wachsender Menge an nicht-professionellen Anwendern, die vergleichsweise geringe Informationskompetenz mitbringen. Zum Vergleich zieht sie die Arbeitsteilung bei der Reparatur eines Autos heran: Die wenigsten Autofahrer möchten lernen, einen Defekt, etwa am Motor, beheben zu können. Statt dessen lassen die meisten den Wagen professionell warten und verlassen sich auf die stets verfügbaren Dienste der Automobilclubs, falls ein akuter Schaden eintritt. Diese Arbeitsteilung ist auch sinnvoll, weil die Technik längst zu komplex geworden ist, um von jedem Anwender erlernt werden zu können.¹

Einigkeit zwischen Maes und Shneiderman besteht in folgendem Grundsatz:

„User understanding is central, and user control is vital for people to be successful.“ (Shneiderman u. Maes 1997, S. 54)

Wenn die Initiative zum Teil von Software-Agenten ausgeht, ist die Kontrolle auf Seiten des Benutzers unter Umständen eingeschränkt. Daher entwickelt sich die Er-

¹ Gerade in der Automobiltechnik ist auch die Automatisierung und Adaptivität von Monitoring und Einstellung von Betriebsparametern weit fortgeschritten so dass hier auch eine etablierte Arbeitsteilung von Mensch und Maschine anzutreffen ist.

forschung von Benutzungsoberflächen mit gemischter Initiative zu einem eigenständigen Forschungsgegenstand, welcher im folgenden Abschnitt beschrieben ist.

3.1.2 Benutzungsschnittstellen mit gemischter Initiative

Ein Anwendungsfall für Agenten sind *Benutzungsschnittstellen mit gemischter Initiative* (MIUI – Mixed Initiative User Interfaces), bei denen die Agentenfunktionen nicht nur auf Abruf oder Initiative des Benutzers zur Verfügung stehen. Der oder die Agenten können bei MIUI-Oberflächen aktiv in das Geschehen eingreifen.

Beispiele sind etwa aktive Hilfesysteme, die die Handlungsstränge der Anwender beobachten und weitere Handlungen vorschlagen, falls Probleme aufzutreten scheinen, oder der Benutzer sein potenzielles Ziel nicht erreicht.

Horvitz (1999) beschreibt einige Grundzüge und Prinzipien solcher Systeme. Anhand des Systems LookOut, einer proaktiven Erweiterung von Microsoft Outlook, beschreibt er die Kernprinzipien und -probleme, die bei solchen MIUI eine Rolle spielen.

- Es wird ungenügend auf die wahren Benutzerbedürfnisse geachtet, oder die Ziele der Benutzer werden falsch eingeschätzt.
- Die Abwägung der Vor- und Nachteile von Automatisierungen werden inadäquat bewertet.
- Die proaktiven Aktionen leiden unter schlechtem Timing.
- Die Möglichkeiten, den Nutzer an die automatisierten Dienste heranzuführen – passend zu seinem Kontext – werden nicht hinreichend genutzt.
- Es wird zu wenig versucht, Problemlösung mit gemischter Initiative zu unterstützen, bei der Nutzer und automatisierte Dienste gemeinsam versuchen, die Ziele des Nutzers zu erreichen.

Horvitz nennt daher zwölf kritische Faktoren, die zur erfolgreichen Integration von direktmanipulativen Oberflächen mit automatisierten, intelligenten Diensten (Agenten) beachtet werden sollten. Diese Kriterien bieten eine sinnvolle Grundlage für die zu erarbeitende Bewertbarkeit der Gestaltung von Schnittstellen mit gemischter Initiative:

3 Arbeitsteilung in agentengestützten Informationssystemen

1. *Die Automatisierung sollte sich auf Dienste mit signifikantem Mehrwert beziehen, die sonst nicht realisierbar wären.*

Automatisierung reicht als Selbstzweck nicht aus. Benutzerautonomie und Kontrolle sind wertvoll und sollten nur zurückgenommen werden, wenn der Mehrwert die Nachteile überwiegt.

2. *Unsicherheit des Benutzers und seiner Ziele müssen einkalkuliert werden.*

Die automatisierten Systemfunktionen könnten Entscheidungen vorwegnehmen, die der Anwender eigentlich selbst treffen müsste. In vielen Bereichen – vor allem in Problemlösungsszenarien – ist es oft so, dass der Benutzer nicht sicher ist, wie sein eigentliches Ziel aussieht und in welche Richtung er arbeiten will. In vielen Fällen können Systeme davon profitieren, diese Unsicherheit einzukalkulieren und Schlüsse daraus zu ziehen. Es sollte versucht werden, die Intentionen und den Fokus der Anwender zu bestimmen.

3. *Die Beschränkungen der Aufmerksamkeitsspanne müssen beim zeitlichen Verhalten der Dienste berücksichtigt werden.*

Die menschliche Aufmerksamkeitsspanne unterliegt Beschränkungen. Das muss beim zeitlichen Verhalten berücksichtigt werden. Gerade das „Timing“ kann ein sehr sensibler Punkt aus Sicht des Anwenders sein. Daher brauchen die Agenten ein gültiges Modell der aktuellen Aufmerksamkeit des Benutzers. Es kann sinnvoll sein, Ergebnisse zu einem späteren Zeitpunkt zu präsentieren, wenn sie dann weniger störend wirken.

4. *Herleitung der idealen Aktion unter Berücksichtigung von Kosten und Nutzen und der Unsicherheit auf Seiten des Anwenders.*

Wenn man davon ausgeht, dass jede Automatisierung auch Kosten erzeugt, indem sie der Benutzerautonomie und Kontrolle abträglich ist oder zusätzlich Zeit kostet, muss der Nutzen diese überwiegen.

5. *Dialoggestaltung und Rückfragen, um Unsicherheit entgegenzuwirken.*

Explizite Dialoggestaltung kann helfen, die Unsicherheit abzubauen. Statt ad-hoc Entscheidungen zu treffen, die die Unsicherheit verstärken könnten, sollte der Benutzer entscheiden können, was zu tun ist. Dies ist gegen die Kosten der Störung und Ablenkung durch Rückfragen abzuwägen

6. *Maßnahmen zum direkten Anstoßen und Abbrechen automatisierter Dienste.*

Es sollte nicht unnötig komplex sein, einen automatisierten Dienst zu starten. Ein autonomes System, welches unter Unsicherheitsbedingungen agiert, wird

manchmal Entscheidungen treffen, die nicht optimal sind. Deshalb sollte der Anwender korrektiv eingreifen und Dienste starten oder beenden können.

7. *Minimierung der Kosten von unzulänglicher Einschätzung der richtigen Aktion oder des richtigen Timings.*

Es sollte versucht werden, den Schaden von falschen Entscheidungen oder von schlechtem Timing zu minimieren. Der Nutzer sollte sich also durch die unpassenden Systemaktionen nicht unangenehm gestört fühlen. Auch sollten die Aktionen nicht zu vordergründig ablaufen oder den Anwender aus seiner eigentlichen Ausgabe herauswerfen.

8. *Skalierbarkeit der Präzision der Dienste in Hinblick auf die Varianz der Unsicherheit der Anwender.*

Der Wert von Automatisierung kann durch die Eigenschaft der Agenten, die Präzision der Funktionen zurückzunehmen und sich so der Unsicherheit der Anwender anzupassen, gesteigert werden.

9. *Mechanismen zur effizienten User-Agent-Kollaboration zur Verfeinerung der Resultate.*

Nutzer wünschen oft, die Analysen der Agenten zu korrigieren oder zu komplettieren. Dies sollte beim Design berücksichtigt werden.

10. *Beachtung der passenden sozialen Verhaltensweisen in der Agent-User-Interaktion.*

Ein Agent sollte ein angemessenes Standard-Verhalten beherrschen, das den sozialen Erwartungen der meisten Anwender entspricht.

11. *Pflege einer Historie der letzten Interaktionen.*

Systeme sollten ein Gedächtnis über die jüngsten Aktionen haben. Benutzer sollten auf einfache und natürliche Weise auf die Objekte und Referenzen der vorangegangenen Ereignisse zurückgreifen können.

12. *Kontinuierliches Lernen durch Beobachtung.*

Automatisierte Dienste sollten die Möglichkeit haben, dazuzulernen und besser zu werden, je länger sie den Anwender und seine Vorlieben kennen.

Während die meisten dieser Punkte allgemeine Gestaltungsrichtlinien in Hinblick auf gemischte Initiative sind, gehen einige (6., 9., 10.) gezielt auf die Mensch-Agent-Kommunikation ein.

Im Folgenden wird ein Überblick über die Konzepte gegeben, die die aktuelle Forschung unter dem Begriff „Agent“ versteht.

3.1.3 Agenten und proaktive, intelligente Systemkomponenten

Jeder Software-Agent ist zunächst ein eigenständiger, proaktiver und stetig laufender Teil eines Softwaresystems. Der Begriff *Intelligenz* ist dabei umstritten, da unklar bleibt, was Intelligenz bei Software im Vergleich zu der des Menschen sein kann.

In der Regel wird *Adaptivität* auf wechselnde Arbeitsumgebungen oder Benutzerziele impliziert. Meist werden weitere Eigenschaften herangezogen, um die Agentenhaftigkeit zu bestimmen. Die genaue Bedeutung des Begriffes *Agent* ist jedoch nicht einheitlich definiert und unterscheidet sich häufig auch in der verwendeten Metaphorik. Es geht dabei, um verteilte Akteure im System, die Aufgaben übernehmen und ausführen können. Die Agenten sollen den Anwendern helfen, ihre Ziele zu erreichen.

Im Hinblick auf Informationssysteme und Information Retrieval argumentiert Kay (1990), dass Anwender nicht voll autonom recherchieren wollen, da die Informationssuche einen hohen Anteil an ermüdenden Routineaufgaben enthält und daher Delegierbarkeit wünschenswert ist:

„A retrieval “tool” won’t do because no one wants to spend hours looking through hundreds of networks with trillions of potential useful items. This is a job for intelligent background processes that can clone their users’ goals and then carry them out. Here we want to indirectly manage agents not directly manipulate objects“

(zitiert nach Etzioni u. Weld (1995)).

Diese Beschreibung passt auf Software-Agenten genauso wie auf jede andere Form von „Intelligenz“ im Softwaresystem. Der Benutzer ist Teil des Systems und andere Benutzer können als Agenten fungieren und unterstützend mithelfen. Agenten können als weitere Benutzer erscheinen, die in einen Dialog mit dem Benutzer treten. Sie können, müssen aber nicht, personifiziert sein. Wichtig ist, dass der Benutzer sich im System nicht als alleinig handelnde Instanz versteht und dass Handlungen aller Beteiligten in seine Lösungsstrategien miteinbezogen werden können. Der Informationsvermittler im Sinne der Bibliothekswissenschaften kann so ein Partner sein, der hilft, das eigentliche Ziel zu erreichen. Auch (Black-/White-)boardsysteme und Wikis oder Informationsablagen sowie persönliche Handbibliotheken können Dreh- und Angelpunkte der Kommunikation zwischen verteilt handelnden Agenten sein.

3.1.3.1 Definitionen

Agenten sind Systemkomponenten, die handeln, um einen Auftrag zu erfüllen oder ein bestimmtes Ziel zu erreichen. Um Agenten gegen andere Systembestandteile abzugrenzen, gibt es eine Reihe von Charakteristika, die auf Agenten angewandt werden (OMG 1999, S. 9, ff.): *Autonomie, Interaktivität, Adaptivität, Mobilität, Proaktivität, Intelligenz, Dauerhaftigkeit, Kooperativität*.

Wie Nwana u. Ndumu (1999) ausführen, wird ein Forschungsgebiet nur durch seine besonderen *Problemstellungen* charakterisiert, nicht aber durch besondere *Methoden* oder *Techniken*. Wenn der Agentenansatz zu sehr ins Zentrum gerückt wird und das Augenmerk zu sehr auf die speziellen Probleme bei diesem Ansatz fällt, geraten die eigentlich zu lösenden Probleme leicht ins Hintertreffen. Deshalb soll bei der Problemanalyse nicht vergessen werden, dass es um die Gestaltung von Informationssystemen geht, die mit Hilfe von proaktiven agentenbasierten Funktionen helfen sollen, Arbeits- und Informationslast zu bewältigen.

Diese Problemstellung war auch zentral in dem Artikel „Agents that reduce work and information overload“ von Maes (1994). Die Idee von Kay aufgreifend, dass hilfreiche Agenten indirekt das Interface zur Informationsrecherche managen und proaktiv Teilprobleme für den Benutzer lösen, die er an die Agenten delegiert hat, entwirft sie eine Metapher: „Persönliche Informationsassistenten“. Beim Beobachten des Anwenders sollen die Agenten lernen, was getan werden muss, oder wie es besser getan werden kann, um dem Benutzer neue Wege aufzuzeigen.

Lernen ist in diesem Kontext eine wichtige Agenteneigenschaft, so wie *den Benutzer beobachten*. Nwana und Ndumu bezeichnen dies als „starke Hypothese über persönliche Agenten“. Im weiteren untersuchen sie diese Hypothese in Hinblick auf ein anderes Szenario: Ein Anwender will sich in einem neuen Betriebssystem zurechtfinden. Der persönliche Assistent (PA) müsste dazu mindestens die folgenden Eigenschaften haben:

- Der PA müsste die typischen Eigenschaften und Eigenheiten des jeweiligen Betriebssystems genau kennen.
- Es wäre wichtig, dass der Agent kontinuierlich ein Modell über den Benutzer pflegt. Ohne ein aktuelles Modell könnte der Assistent keine personalisierten Hilfestellungen geben.
- Der PA brauchte ein klares Verständnis und ein gültiges Workflow-Modell für die Aktionen, die der Benutzer im Interface tätigt. Er müsste in der Lage sein,

die jeweils aktuelle Intention des Benutzers herzuleiten. Das würde ein tiefgehendes kognitives Modell des Anwenders zur Bedingung machen.

Daraus folgt, die von Maes implizierte Aufgabe des intelligenten Lehrens (*Tutoring*) und Leitens (*Mentoring*) kann der PA nur mit vier Wissensbasen bewältigen:

1. Wissen über die Anwendungsdomäne
2. Ein Modell des Benutzers
3. Strategien für die mögliche Assistenz
4. Einen Katalog typischer Probleme und passender Lösungen im Anwendungsbereich

Nwana bemerkt, dass Maes starke Hypothese angenommen hat, das Problem des *intelligent tutoring* sei bereits gelöst. Da das nicht der Fall war, auch nicht bis einschließlich 2006, seien weit schwächere Hypothesen an deren Stelle getreten.

Außerdem weist Nwana darauf hin, dass die ersten Versuche, diese Ansätze zu implementieren, zu hitzigen Debatten geführt haben, wann es wirklich notwendig ist, solche Assistenten anzubieten.

Es gibt somit einige spezifischere Definitionen von Agenten, die in Informationssystemen Verwendung finden und auf weniger starken Hypothesen aufbauen, als der PA von Maes:

Information Agent: „Intelligente *Information Agents* haben Zugang zu verschiedenen, heterogenen und geographisch verteilten, Informationsquellen im Internet oder in firmeneigenen Intranets. Die Hauptaufgabe solcher Agenten ist die proaktive Suche für den Benutzer, die Verwaltung und Mediation von Information für Benutzer und andere Agenten.“ (Klusch 2001, S. 2)

Interface Agent: Agenten, die die Aktionen des Anwenders beobachten und auf bestimmte Ereignisse reagieren. Sie können Teile der Schnittstelle steuern oder umkonfigurieren (Adaptivität), Hilfestellung geben oder im Dialog mit dem Anwender Probleme lösen (*Assistenten, Wizards*). (Bradshaw 1997; Maes 1994; Lieberman 1997)

Während bei den *Information Agents* die indirekte Kommunikation, die Delegation und die Ablaufkontrolle im Vordergrund stehen, geht es bei den *Interface Agents* vorrangig um Dialoge mit den Agenten und um Dialogführung und Ausgestaltung.

3.2 Mentale und kognitive Modelle

Um die potentiellen Konflikte zwischen Anwendern und Agenten behandeln zu können, muss man verstehen, welche kognitiven Modelle in diesem Kontext eine Rolle spielen. Die kognitiven Modelle werden entscheidend durch die verwendeten Metaphern geprägt. Erwartungen und Befürchtungen bestimmen ebenfalls das Verhältnis zwischen Anwendern und Agenten.

3.2.1 Metaphern

Metaphern sind sowohl bei der Gestaltung von Benutzungsschnittstellen als auch bei der Interpretation einer Anwendung auf Seiten des Benutzers sehr bedeutsam. Durch sie werden Erwartungshaltungen gegenüber dem Verhalten der Software und ihres Funktionsumfangs geprägt. Zudem wird einiges an assoziiertem Weltwissen transportiert.

Persönlicher Assistent Negroponte (1997) beschreibt als Beispiel die Rolle des *englischen Butlers*, der sich um die Belange seines Dienstherrn kümmert und Telefonanrufe entgegen nimmt, die Anrufer erkennt und nur stört, wenn es angebracht erscheint. Dieser Agent ist gut trainiert in Bezug auf Timing, versiert wenn es darum geht, die angebrachten Momente zu erkennen und respektvoll gegenüber Eigenheiten seines Gegenübers.

Im Alltagsleben weiter verbreitet als Butler (und somit metaphorisch aussagekräftiger – viele Anwender wüssten sicher nicht auf Anhieb, wie sie sich ihrem Butler gegenüber verhalten sollten) wäre die Rolle einer Sekretärin im Büroumfeld, die einen gut kennt und Zugriff zu relevanten Informationsquellen hat, etwa Adressregister oder Kalender (Termine). Solche Personen können auf effektive Weise in unserem Interesse handeln.

Wie Negroponte bemerkt, ist so ein gut trainierter Agent nicht einfach zu ersetzen. Selbst ein hochintelligenter Ersatz („Albert Einstein“) könnte das trainierte Spezialwissen nicht ersetzen.

Vermittler Ein Informationsvermittler (Mediator) (vgl. Ingwersen 1992) oder -Broker könnte die klassische Schnittstelle zwischen Konsument und Informationsanbieter besetzen. Bibliothekare und Informationsvermittler gibt es schon lange,

das Internet hat jedoch die eigenständige Recherche in den Vordergrund gerückt. Aber die Informationskompetenz bei professionellen Vermittlern ist in der Regel weit höher und somit sind Recherchen effektiver. Das Rollenbild wäre somit durch einschlägiges Expertenwissen über Informationsquellen, Strategien und Taktiken sowie Kenntnis über Anfragesprachen und andere Eigenheiten der heterogenen Zugänge geprägt.

Ratgeber Negroponte nennt das Rollenbild der *virtuellen Schwägerin*, die zum einen über ausgeprägtes Wissen über das aktuelle Kinoprogramm besitzt und zum anderen sehr gut die eigenen Vorlieben kennt. Viele Leute verlassen sich gerne auf das Urteil solcher persönlicher Ratgeber. Auch hier ist relevant, dass der Ratgeber ein gutes und passendes Modell der eigenen Vorlieben hat.

Etzioni u. Weld (1995) charakterisieren Agenten anhand der Datenautobahn-Metapher. Sie gehen von einem Gedankenexperiment aus, bei dem die Problemlösung damit verknüpft ist, auf der „Datenautobahn“ an einen anderen Ort zu gelangen. Dabei könnten die folgenden Typen von Agenten hilfreich sein:

Fahrlehrer oder Beifahrer, die an jeder Abzweigung Richtungshinweise geben. Diese Metapher liegt zum Beispiel auch den für PKW mittlerweile gebräuchlichen Navigationssystemen zugrunde. Die Initiative bleibt beim Benutzer.

Ein Taxifahrer, der einen an das genannte Ziel fährt. Die Initiative liegt nur beim Beginn der Fahrt beim Benutzer, danach handelt der Agent weitgehend autonom, während der Benutzer passiv bleibt.

Ein Concierge, der aufgrund seines gesammelten Wissens und seiner Fähigkeiten die ganze Autofahrt überflüssig werden lässt. Hier ist durch die besonderen Fähigkeiten ein Mehrwert entstanden, der einen komplett anderen Arbeitsfluß ermöglicht.

Die Dimensionen, die in diesen Metaphern variiert wurden, stellen die *Handlungsautonomie des Benutzers*, sowie die *Funktionsfülle* und *Intelligenz des Agenten* dar.

In Hinblick auf Informationssysteme wird von Bates (1990) argumentiert, dass zu viel Automatisierung gerade in Problemlösungssituationen wie dem Information Retrieval kontraproduktiv sein kann und die Hauptentwicklungsarbeit bei Systemen mit gemischter Initiative und ausgewogener Arbeitslast liegen sollte.

Umgekehrt betrachtet, lässt sich auch von variabler *Autonomie der Agenten* sprechen, wie dies bei Scerri (2001) definiert wird.

In ähnlicher Weise definiert Belkin „Intelligentes Information Retrieval“ als *System mit verteilter Intelligenz, zwischen Nutzer und System*. (Belkin 1996)

3.2.2 Kooperationsmodelle

Bei der Arbeitsteilung geht es auch oftmals nicht alleine um die Rollen, die die Beteiligten einnehmen, sondern auch um die Kooperation und die sinnvolle Aufteilung der anstehenden Aufgaben. Im Folgenden sind Aspekte und Kooperationsmodelle beschrieben.

Delegation Der am häufigsten genannte Fall eines Kooperationsmodells mit Agenten ist das Delegieren von Aufgaben: Der Benutzer spezifiziert eine Arbeitsaufgabe oder ein zu erreichendes Ziel. Der Agent erledigt die Teilaufgabe und liefert die Ergebnisse zu einem späteren Zeitpunkt. In der Zwischenzeit möchte der Auftraggeber eventuell kontrollieren, wie weit die Aufgabe erledigt werden konnte und auch eventuell die Bearbeitung abbrechen lassen. Der Agent handelt jedoch autonom und darf eigenständig Entscheidungen treffen. Es muss also bei den Arbeitsprozessen des Agenten einen Monitoring-Mechanismus geben, der Einblick in Fortschritt und Entscheidungsprozesse gewährt.

Mentoring Beim Mentoring begleitet ein erfahrener Agent den nicht so erfahrenen Benutzer, leitet ihn teilweise und gibt Hilfestellungen. Er kann auch bei Fragen zur Herangehensweise oder Strategie konsultiert werden. Der Anwender handelt weitgehend autonom. Der Mentor darf aber eingreifen, wenn das Vorgehen nicht optimal ist.

Collaborative Coaching Beim Collaborative Coaching (vgl. Brajnik u. a. 2002) sind beide Partner nur teilweise im Besitz des zur Problemlösung notwendigen Wissens. Durch die gemeinsame Herangehensweise können sie gegenseitig ihr Halbwissen auffüllen und so das Problem lösen. Beide Partner handeln weitgehend autonom und gleichberechtigt.

3.2.3 Psychologische Aspekte

Bei der Interaktion mit Software-Agenten spielen psychologische Aspekte eine starke Rolle. Der Agent ist im sozialen Kontext zu sehen. Wie wird er wahrgenommen, was erwartet der Benutzer und was befürchtet er?

Norman (1994) weist darauf hin, dass die den Agenten zugeschriebenen Attribute, wie „Intelligenz“, aber auch schon das Konzept eines „Agenten“ *menschliche* Eigenschaften erwarten lassen. Wie interagieren Mensch und Agent und wie denken Menschen über Agenten? Norman argumentiert, dass Automatisierung und autonome Komponenten kein ganz neues Konzept sind. Servomechanismen, wie die Autopiloten von Flugzeugen oder die automatisierte Fabriksteuerung gibt es schon lange. Software-Agenten haben allerdings das Potential, komplexer zu sein, als diese Servomechanismen sein können und haben die Möglichkeit, in Dialog mit dem Anwender zu treten. Der Gestaltungsspielraum umfasst auch natürlichsprachliche Schnittstellen, Grafik mit Gestik und Mimik usw. Software-Agenten können eigene Ziele entwickeln und Handlungspläne entwerfen und verfolgen. Dies weckt höhere Erwartungen seitens des Benutzers, aber auch Befürchtungen, die Agenten könnten nicht in seinem Sinne handeln.

Kann man die Schnittstelle so gestalten, dass die induzierten Erwartungen nicht die Möglichkeiten der Agenten übersteigen und dass Befürchtungen abgebaut und Vertrauen in die Agenten aufgebaut werden kann?

Die Erwartungshaltung der Anwender wird durch die Repräsentation der Agenten auf der Benutzungsoberfläche entscheidend geprägt. Die gewählten Metaphern und die Gestaltung der Dialoge und der gesamten Schnittstelle zu den Agenten sollte möglichst präzise die tatsächlichen Fähigkeiten und Eigenschaften der Agenten spiegeln.

Es gibt beispielsweise einen Disput über die Repräsentation von Agenten in Form von anthropomorphen Charakteren. Die Kritiker meinen, dies sei irreführend und würde zu hohe Erwartungen an Intelligenz und Dialogfähigkeit der Agenten wecken. Statt dem Anwender „lebensechte“ Charaktere vorzugaukeln, sei es wichtiger, die bestehende Funktionalität in anwendungsgerechter Form anzubieten. Anthropomorphe Gestaltungen verlangen wenigstens so viel Intelligenz vom Agenten, dass er sozial sinnvolles Verhalten beherrscht, welches der jeweiligen Situation angemessen ist. Diese muss analysiert werden.

Das PERSONA-Projekt von Microsoft Research (Ball u. a. 1997) versuchte, anhand der Event-Analyse mittels Bayesischer Netze solche Aussagen zu treffen. Die umstrittene Implementation als „hilfreiche“ Büroklammer wurde als zu aufdringlich und nicht treffsicher kritisiert. Horvitz blickt heute selbst kritisch auf diese Implementierung zurück (Markoff 2000). Dennoch bleibt die Frage, ob menschenähnliches Verhalten

und das damit eingeführte „soziale Verhalten“ eine günstige Sichtweise auf Agenten ist. Die Anwender kommunizieren dann mit dem Agenten, statt Werkzeuge direkt manipulativ für ihre Problemlösung zu benutzen.

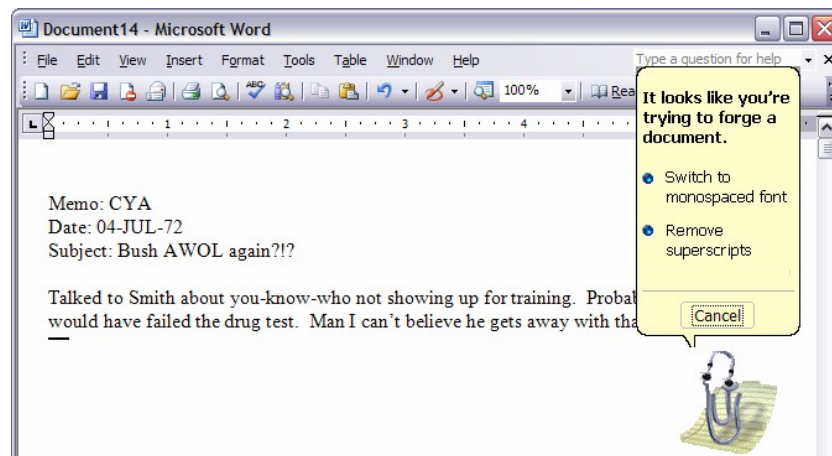


Abbildung 3.1: Assistent für Microsoft Word, der Vorschläge zum aktuellen Kontext anbietet.

Die Befürworter anthropomorpher Gestaltung führen an, dass die Dialogführung, wenn sie der jeweiligen Situation angemessen gestaltet wurde, leicht erlernbar ist und dass mit Gestik und Mimik der Charaktere eine zusätzliche emotive Ausdrucksdimension in vielen Situationen hilfreich sein kann. Außerdem lassen sich solche personifizierten Agenten auch als Ratgeber und „Reiseleiter“ in Präsentationen einsetzen. Die Hoffnung ist, die Hürde für Anfänger zu senken und einen Unterhaltungswert einzuführen. (Rist u. a. 1997)

In einem Artikel bei Telepolis² wird diese Sichtweise so formuliert:

Als in den 1980er Jahren die automatisierte Textverarbeitung in den Büros Einzug hielt, trat bald ein seltsames Phänomen zutage, das Psychologen später als den Chefsekretärinnen-Effekt bezeichneten: Auch wenn Arbeitsabläufe über Sekretärinnen länger dauerten, weigerten sich viele Kräfte, auf ihre Bürokraft zu verzichten und ihren Schriftverkehr selbst zu tätigen.

Anfangs wurde dies auf Berührungsängste älterer Mitarbeiter mit der Datenverarbeitungstechnologie zurückgeführt. Bei näheren Untersuchungen

²Telepolis Artikel vom 1.4.2007-URL:

<http://www.heise.de/tp/r4/artikel/24/24963/1.html>

stellte man jedoch fest, dass auch jüngere und mit den Grundlagen elektronischer Datenverarbeitung vertraute Sachbearbeiter und Führungskräfte ungern auf ihre Sekretärinnen verzichteten. Ursache war zum einen ein Machtverlust in der Firmenhierarchie - eine Sekretärin war weiterhin ein Statussymbol - zum anderen gibt es aber auch ein menschliches Grundbedürfnis nach möglichst individuellen Befehlsempfängern. Es macht, so fanden die Psychologen heraus, mehr Spaß, einer Arbeitskraft eine Aufgabe zu delegieren als diese Aufgabe durch ein technisches und relativ unsichtbares Hilfsmittel zu erledigen.

Auf die Erkenntnis des Vorhandenseins dieser anthropologischen Konstante stützt sich ein neues Vorhaben von Yahoo!: Die Verbindung von effizienter und billiger Technologie mit allen menschlichen Bedürfnissen gerechter werdender Personalisierung. Die Entwicklung von Suchrobotern (*Knowbots*) auf der Basis von bestehender Suchmaschinentechnologie, aber in personalisierter Form. Die wie Cartoon-Figuren gestalteten Test-Suchroboter *Bob*, *James* und *Ramigani* sind als Roboter, englische Butler und indischer Lakai gestaltet.

Die anthropomorphen Knowbots sollen nicht nur neue Marktsegmente erschließen, sondern auch Umsteiger von Google anlocken. Gerade Aufgaben, wie der Kauf von Produkten mit Preisvergleich oder die Bestellung von Dienstleistungen, sind für viele Menschen angenehmer erledigt, wenn sie diese abends einem Lakaien auftragen, der sie am nächsten Morgen mit der erledigten Aufgabe überrascht, sich bedankt und verabschiedet. „Lassen sie ihr Geld für sich arbeiten“ war nicht zufällig einer der erfolgreichsten Slogans, mit denen Banken in den letzten Jahrzehnten für sich warben.

Das komplizierte Timing und die grundsätzlich verschiedene Modalität von Kommunikation versus direkter Manipulation macht die Realisierung jedoch schwierig. Hinzu kommt, dass die *Aufmerksamkeit* als menschliche Eigenschaft bei der Arbeit am Computer bisher noch wenig erforscht ist. Dieses Problem wird beispielsweise im „Attentional User Interface (AUI) Projekt“ (Horvitz u. a. 2003) erforscht. Die Benutzeraufmerksamkeit ist beschränkt und bei konzentrationsintensiven Aufgaben reagieren Benutzer sehr empfindlich auf Störungen.

Forschungen zur menschlichen Kognition (Erikson u. Yeh 1985; Gillie u. Broadbent

1989) zeigen auf, dass es Schranken gibt, wie viele Dinge man bewusst wahrnehmen oder behalten kann und wie viele Handlungsstränge man gleichzeitig verfolgen kann. Auch die Fähigkeit, unterbrochene Handlungen wieder aufzunehmen, unterliegt gewissen Schranken. Beispielsweise wird die Wiederaufnahme wahrscheinlicher, wenn man dem zu erreichenden Ziel näher war. An unterbrochene Tätigkeiten kann man sich in der Regel besser erinnern, als an abgeschlossene.

Die Resultate der kognitionswissenschaftlichen Forschung belegen insgesamt, dass Menschen mit ihren kognitiven Begrenzungen nicht umgehen können. Zu den Grundlagen zwischenmenschlicher Kommunikation gehört deshalb auch ganz zentral das Erzeugen und Beachten von Signalen, die Aufmerksamkeit, Zuhörbereitschaft und Turn-Taking-Möglichkeiten anzeigen.

Die Beachtung solcher Signale ermöglicht flüssige Kommunikation und Zusammenarbeit (Horvitz u. a. 2003). Die Nichtbeachtung hingegen wird als unhöflich und störend oder „nervig“ empfunden. Da Software-Agenten in der Regel unzureichende Sensorik besitzen, um die meist unbewusst und non-verbal erzeugten Signale der Benutzer zu erkennen und zu deuten, können sie nur unter groben Annahmen mit dem Benutzer in Kontakt treten – meist im „unpassenden“ Moment.

Das oben erwähnte AUI Projekt versucht durch erweiterte Mechanismen (Gestik- und Mimikerkennung) und präzisere Modellbildung, unter Berücksichtigung von Kosten und Nutzen der Verschiebung von Benachrichtigungen, solche Facetten der Kommunikation einzubeziehen, um Benutzerschnittstellen mit gemischter Initiative wirksamer zu machen. Eine andere Möglichkeit ist die indirekte Beobachtung des Benutzers durch Auswertung seiner persönlichen Datenspeicher, wie z.B. sein Kalender oder die Dokumentablage. So lassen sich Wahrscheinlichkeiten für die potentielle Nützlichkeit neuer Nachrichten und Information bestimmen. Die Wahrscheinlichkeit für die potentiellen Kosten (Störung) müssen dann ebenfalls anhand eines Benutzermodells berechnet werden, das auf indirekten Beobachtungen und Modellbildung aufbaut.

Die Vermutung, dass anthropomorphe Schnittstellen zu höherer Anwenderzufriedenheit beitragen, ist aber unter den derzeit gegebenen Einschränkungen der Mensch-Computer-Interaktion nicht schlüssig.

Eine andere zentrale psychologische Dimension entfaltet sich zwischen Kontrolle und Autonomie (Friedman u. Nissenbaum 1997; Friedman 1998; Norman 1994). Autonomie bedeutet hier, dass der Anwender die Möglichkeit behält, selbst-bestimmt

auf sein Ziel hinzuarbeiten. Wenn er dabei die Hilfe von Software-Agenten in Anspruch nimmt, will er deshalb nicht gänzlich die Kontrolle abgeben. Ungünstige Gestaltung, gerade in Hinblick auf Agenten, kann aber zu empfindlichen Verlusten der Kontrolle führen; aber auch grundsätzliche Unterschiede zwischen menschlichen Kommunikationspartnern und Software-Agenten können diesbezüglich ernsthafte Problemen hervorrufen. Um dies zu illustrieren, führen Friedman und Nissenbaum den Vergleich zu menschlichen Assistenten an: Zum einen würde in zwischenmenschlichen Konstellationen zwischen einem Handelnden und einem Assistierenden die Möglichkeit bestehen, Einschränkungen der Autonomie zu kommunizieren und diese somit wieder herzustellen. Zum anderen wäre ein Assistent, der nicht gut assistiert, ein spezieller, an eine bestimmte Person gebundener, Einzelfall. Schlecht gestaltete Software-Agenten würden hingegen potentiell tausenden von Nutzern Schwierigkeiten bereiten.

Darüber hinaus lassen sich auch Konsequenzen für die Verantwortung für die eigenen Handlungen ableiten: Wenn die Autonomie einer Person gewahrt bleibt, kann man ihm auch die Verantwortung für sein Handeln anlasten. Ist seine Autonomie eingeschränkt, muss er sich ggf. nicht mehr voll verantworten.

Norman (1994) illustriert das Autonomieproblem anhand von servomechanischen Einrichtungen, wie den Autopiloten in Flugzeugen. Obwohl es technisch machbar wäre, den Autopiloten ohne menschlichen Piloten fliegen zu lassen, bleibt die Autonomie der menschlichen Piloten eine wichtige Regel im Flugverkehr. Das Vertrauen in menschliche Akteure ist begründeterweise immer größer als in technische Lösungen.

Berühmt ist auch das Ringen um Autonomie in dem Spielfilm „2001 – Odyssee im Weltall“: Die Akteure, Mensch gegen intelligenten Bordcomputer, versuchen, sich gegenseitig „auszuschalten“, um jeweils volle Autonomie bei der Steuerung eines Raumschiffes zu erhalten.

Insgesamt ist die Autonomiedebatte nah an der Angst und Mißtrauensgrenze gegenüber der Agententechnologie. Vertrauensvolle Zusammenarbeit setzt eine intakte Autonomie des Benutzers voraus. Daher ist es besonders wichtig, auf eine Lösung dieses Konflikts zu achten.

Insbesondere in Bezug auf die Rechtesituation des Nutzers in Informationssystemen gibt es jedoch einen nicht zu unterschätzenden Aspekt, wenn es um die Autonomie geht: Den unsicheren Zustand am Beginn einer iterativen Recherche, den *Anoma-*

lous State of Knowledge (ASK) Belkin (2000). Nach Belkins Erkenntnis ist die Formulierung einer Anfrage mit so viel Unsicherheit behaftet, dass dies oft zum Scheitern der ganzen Recherche führen kann.

Das System sollte demnach nicht einfach auf eine Anfrage warten, sondern möglichst helfen herauszubekommen, was das eigentliche Informationsbedürfnis des Nutzers ist. In diesem Fall werden vom System angebotene Hilfestellungen und Hinweise vermutlich nicht unbedingt als Verlust der Autonomie wahrgenommen, sondern können aus ausweglosen Situationen heraus helfen.

Vertrauen bzw. Misstrauen ist ein kritischer Punkt im Verhältnis der Nutzer zu den Agenten. Den Agentenkriterien zufolge sind die meisten Viren, Würmer und Trojaner schädliche Vertreter der autonom agierenden Software, die Skepsis gegenüber solcher Software ist also groß. Um so wichtiger ist eine durchgehende Transparenz der Agententätigkeiten und der jeweiligen Arbeitsergebnisse. Das Bedürfnis der direkten Kontrolle und der Überwachbarkeit aller Prozesse sinkt erst, wenn das Vertrauen in die Ergebnisse der Automatisierung sehr hoch und stabil ist. Vertrauensbildung und Zuverlässigkeit sind also unabdingbar.

3.3 Einsatz von Agenten in Informationssystemen

Nach den Betrachtungen über allgemeine Anforderungen an Agenten werden im Folgenden agentenbezogene Konzepte, Beispiele und Entwurfsmuster diskutiert, die im Zusammenhang mit Informationssystemen stehen.

3.3.1 Information Agents

Die *Information Agents* sind eine Klasse von Agenten, die bei der Informationssuche und beim Filtern von Informationsströmen helfen. Auf der einen Seite sollen sie helfen, Suchfunktionen auf höherer Ebene zu ermöglichen, auf der anderen Seite sollen sie die Informationsvermittlung personalisieren helfen und effizienter machen. *Information Agents* und die zugehörigen Konzepte werden bei Klusch (1999; 2001) zusammengefasst.

Spezielle Unterklassen sind die

- *Information Broker*
- *Monitoring Agents*
- *Just in time IR Agents (JITIR)*

Diese werden in den folgenden Abschnitten näher betrachtet.

3.3.1.1 Information Broker

Die *Information Broker*, dt. *Informationsvermittler-Agenten*, nehmen Anfragen des Benutzers entgegen und liefern aus einer oder mehreren Quellen relevante Information zu dieser Anfrage. Sie entsprechen der Rolle eines *Mediators*, die man aus den Bibliotheks- und Informationswissenschaften kennt. Der Vermittler hat dabei die Rolle, ein Modell über den Fragenden und sein Hauptinteresse aufzubauen und es mit seinem Wissen über die Suchdomäne abzugleichen. Er kann daraufhin direkt die Information beschaffen oder indirekt auf die erfolgversprechenden Wissensquellen oder Suchstrategien hinweisen. Ingwersen (1992) beschreibt ein Interaktionsmodell

für Information Retrieval, das *Mediator Model*, in dem folgende Funktionen auch für Informationsvermittler-Agenten charakteristisch sind:

Funktionen	Beschreibung
Domänenmodell	Enthält gängige Arbeitsschritte (<i>work tasks</i>) der Domäne, Haupt-Themengebiete, Paradigmen und <i>Conceptual Maps</i> . ³
Systemmodell	Enthält die <i>Systemeinstellungen</i> , Informationsquellen und Dienste, die für die Domäne wichtig sind, <i>IR-Techniken</i> für die Suche; <i>Regeln</i> für die Repräsentation der Daten.
Benutzermodell	Enthält das generelle Wissen über <i>Suchverhalten</i> , <i>Präferenzen</i> , <i>Werte und Erwartungen</i> sowie <i>Intentionen</i> in Relation zu den Aufgaben in den Domänen, sowie über den <i>Anwenderstatus</i> und <i>Level</i> .
Systemmodell-Adapter	Wissen über die Strukturen der angeschlossenen Datenbanken; lernt die Spezifika für das Systemmodell.
Benutzermodell-Erzeuger	Generiert analytisches Wissen über die <i>tatsächlichen Nutzercharakteristiken</i> , basierend auf den Attributen im Benutzermodell.
Retrieval-Strategie	Wählt die passende Information-Retrieval-Strategie und führt sie aus.
Antwort-Generierer	Bestimmt die passende Reaktion auf die Anwendersituation. Dies kann <i>Feedback</i> , <i>Information-Transformation</i> oder <i>Anpassung des Benutzermodells</i> sein.
Feedback-Generierer	Generiert an die Situation angepasstes Feedback.
Anfragemodell-Erzeuger	Generiert analytisches Wissen über <i>verfügbare Information</i> und <i>Grundproblem des Anwenders</i> in Form einer Ontologie.
Abbildungen-Erzeuger	Generiert Abbildungen zwischen Inhalten des Anfragemodells und Funktionen, speichert sie und frischt sie auf. Dabei werden individuelle Benutzermodelle und Ontologien berücksichtigt.

³inzwischen würde man dies eine *Ontologie* nennen.

3 Arbeitsteilung in agentengestützten Informationssystemen

Erklärung	Beschreibt Mechanismen und Operationen des zugrundeliegenden Retrievalsystems, abhängig von Benutzermodell und Domäne.
Umformer	Legt den Dialogmodus fest, abhängig vom Benutzermodell und von den erzeugten Abbildungen. Konvertiert Eingaben der Anwender genauso wie Ausgaben der Information-Retrieval-Systeme.
Planer	Prozessiert Regeln für alle anderen Funktionen für Intentionen, Erwartungen und Werte.

Nach diesem Modell gilt, dass die Mediation zwischen Angebot und Suchendem tiefes und dynamisch angepasstes Wissen über den Suchenden und auch die Domäne benötigt. Viele dieser Komponenten sind bis dato hypothetisch, weil zugrundeliegende Probleme noch ungelöst sind. So fehlen brauchbare Domänen- und Benutzermodelle in generalisierter Form, die als Startpunkt dienen könnten, denn auch bevor die suchende Person dem System vertraut ist, soll dieses ja schon brauchbare Empfehlungen und Informationen liefern. Ein Teil dieses Problems, die formale Repräsentation dieser grundlegenden Modelle, wird von der semantischen Modellierung in Form von Ontologien zu lösen versucht.

Auch andere Aspekte, wie zum Beispiel die Regeln über Erwartungen und die genauen Lernmechanismen, sind noch nicht direkt umsetzbar. Somit muss dieses Modell, auch wenn es sich um Konkretisierung bemüht, als abstraktes Modell gesehen werden.

Dennoch wird die Problematik der informationsvermittelnden Schnittstelle gut charakterisiert. Die Benennung der Komponenten und der Bezug zum wirklichen Mediator implizieren auch die Agentenhaftigkeit und Proaktivität des Systems. Da die Informationskompetenz des einzelnen Suchenden, insbesondere auch in Problemsituationen gestärkt werden soll, ist eine kontextsensitive, lernende und adaptive Herangehensweise unvermeidbar.

Eine Methode, Adaptivität zu erreichen, wird durch die Anwendung von Kostenmodellen und Kostenoptimierung sowie Verhandlungsprotokollen zwischen den teilnehmenden Agenten in einem Multi-Agenten-System erreicht. Nach der Berechnung

der optimalen Kosten-Nutzen-Relation kann so dynamisch entschieden werden, welche Information letztlich von welchen Agenten übermittelt, beziehungsweise angeboten, werden soll.

3.3.1.2 Monitoring Agents

Neben der direkten Vermittlung von Information ist es bei der Recherche und damit verknüpften Tätigkeiten⁴ wichtig, die im Hintergrund ablaufenden Prozesse zu beobachten. Agenten, die zur Beobachtung von etwas angesetzt werden, nennt man *Monitoring Agents*. Sie können sowohl zeitnahe Prozesse beobachten, wie etwa die Tätigkeiten während der aktuellen Recherche, als auch länger andauernde Prozesse, wie die Veröffentlichungen oder Neuigkeiten in einer bestimmten Domäne. Besonders das Einrichten von Suchagenten für sogenannte *stehende Anfragen* ist ein beliebtes Merkmal eines modernen Informationssystems. Beispielsweise findet man dieses Feature bei Job-Börsen. Auch für andere Anwendungsbereiche wie die Betriebssystem- oder Netzwerkkontrolle sind diese Art von Agenten von steigender Bedeutung. Um den Benutzer, oder allgemeiner: ihre Klienten, von Neuigkeiten in Kenntnis zu setzen, benötigen die Monitoring Agents einen Informationskanal zum Interessenten, in der Regel zum Anwender. Das kann in Form von E-Mails sein, oder als RSS-Feed, oder abrufbaren Webseiten – auch Popup-Notifications sind gängig.

3.3.1.3 Just in Time Information Retrieval Agents

Just in Time Information Retrieval Agents sind von Rhodes u. Maes (2000) beschrieben worden. Sie beobachten einen Arbeitsvorgang des Anwenders, etwa das Erstellen von Texten in einem Editor, oder das Browsen von Web-Seiten und recherchieren zu diesem Kontext parallel in weiteren Datenquellen.

Zwei Beispielimplementationen werden bei Rhodes beschrieben:

Der *Remembrance-Agent* ist ein Plug-In für den Text-Editor Emacs. Während der Eingabe von Text im Emacs-Textpuffer wird zum Kontext passende Rechercheergebnisse in einem weiteren Puffer präsentiert. *Letizia* ist eine Web-Browser-Ergänzung, die kontinuierlich und autonom die Webseiten-Abrufe des Nutzers beobachtet und inhaltlich passende Seiten und Links vorschlägt.

⁴vgl. auch den Abschnitt 4.2.2 zur Modellierung von Rechercheoperationen in Daffodil.

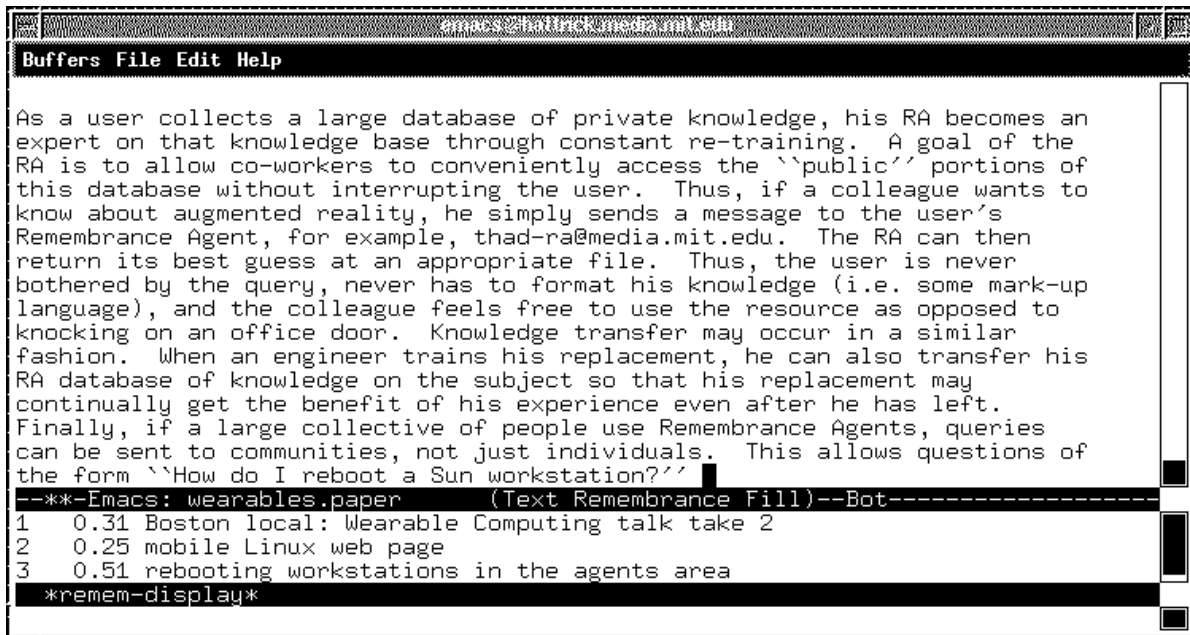


Abbildung 3.2: Der Remembrance Agent

Eine Abwandlung von dieser Idee fand sich zeitweise auch in der „What's related“-Funktion des Netscape Browsers. Dort wurde ein Web-Service der Firma Alexa Internet benutzt, um Seiten in einer Seitenleiste aufzulisten, die inhaltlich zu der Seite passen, die der Anwender gerade geladen hat.

Diese Funktion ist jedoch durchaus kontrovers diskutiert worden. In einem Artikel formulieren Curtin u. a. (1998) folgende recht generelle Kritik, an solchen Funktionen, die nicht lokal arbeiten, sondern als Web-Service an zentraler Stelle Daten sammeln:

“The best-intended systems can sometimes have undesirable consequences. For example, if Netscape were to be purchased by a larger organization that does not respect its customers' privacy, the data that Netscape has collected would then be in "their" hands. Imagine detailed dossiers, including the names of the users, of web users around the world being sold to marketers. Or, perhaps significant changes in Netscape's fortunes will cause it to reconsider its stand on what information it will sell to third parties, if someone is offering enough money for the data, and will guarantee deniability.”

Die Idee bei dieser Herangehensweise ist es, den Anwender von der kognitiven Last der eigentlichen Recherche zu befreien und stattdessen einem Agenten diese Tätig-

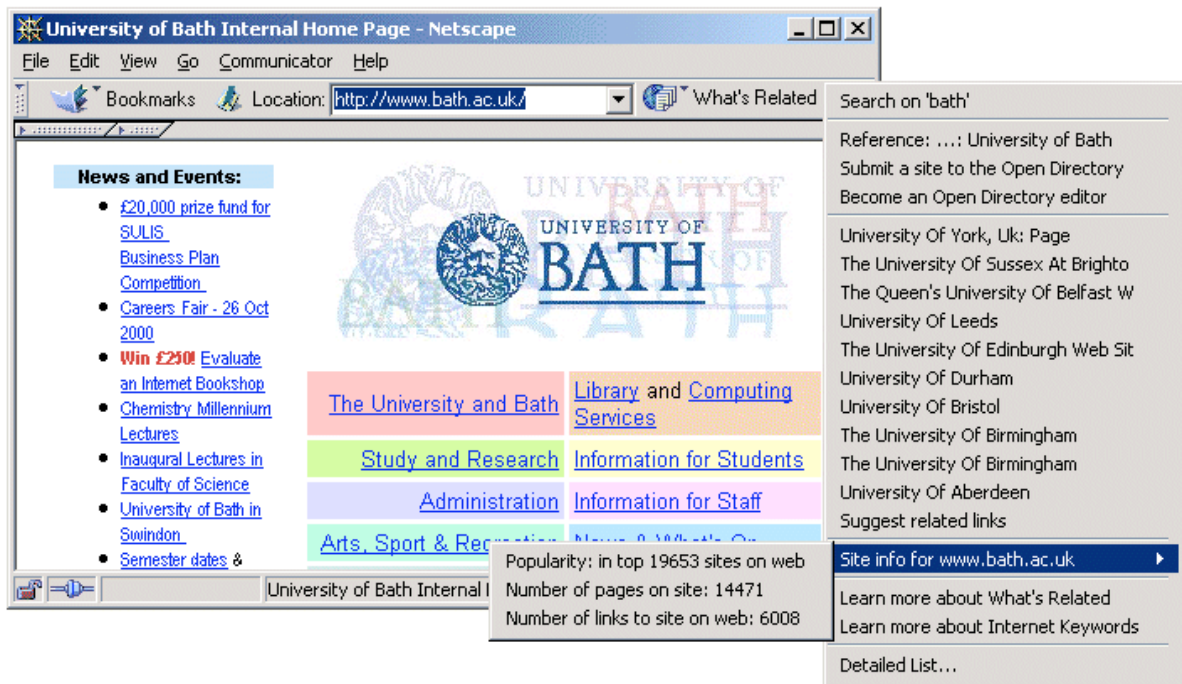


Abbildung 3.3: Netscapes What's related

keit zu überlassen. Der Agent läuft kontinuierlich nebenher und beobachtet den Kontext und die Tätigkeit des Nutzers. Wenn aus dem Index, der dem Agenten zur Verfügung steht, Inhalte zum Anwenderkontext passen, werden sie angezeigt, so als hätte der Anwender danach gesucht.

Schon Vannevar Bush beschreibt solche assoziativ arbeitenden Retrieval-Mechanismen als möglichen Mechanismus gegen die Überlastung durch die Informationsflut:

"Man cannot hope fully to duplicate this mental process artificially, but he certainly ought to be able to learn from it. In minor ways he may even improve, for his records have relative permanency. The first idea, however, to be drawn from the analogy concerns selection. Selection by association, rather than indexing, may yet be mechanized. One cannot hope thus to equal the speed and flexibility with which the mind follows an associative trail, but it should be possible to beat the mind decisively in regard to the permanence and clarity of the items resurrected from storage. "

(Bush 1949, S. 6)

Die Integration von Suchmaschinentechnologie in zeitgenössische Betriebssysteme, wie bereits bei Apples Mac OS X 10.4 in Form von Spotlight, in Linux durch Beagle

und in Windows durch diverse Produkte wie beispielsweise Googles Desktop Suche, der X1/Yahoo Search und die in Windows Vista integrierte Suchmaschine, macht solche nebenläufigen und kontextbezogenen Suchen möglich.

Die *Just in Time Information Retrieval*-Funktion wird auch in sofern vielversprechend, als es im Desktop-Kontext für Agenten viel eher möglich ist, ein persönliches Interessenprofil zu erlernen. Dabei wird die Privatsphäre des Anwenders gewahrt, weil keine persönlichen Daten an eine zentrale Stelle gemeldet werden müssen. Die Informationen zum Benutzerprofil müssen seinen Rechner nicht verlassen, um zu funktionieren.

3.3.2 Interface Agents

Interface Agents sind eine wichtige Gruppe von Software-Agenten, neben den Information Agents, welche besonders für die Information-Retrieval-Unterstützung in Informationssystemen eingesetzt werden kann. Insbesondere das Massachusetts Institute of Technology (MIT) hat sich mit Interface Agents auseinandergesetzt. Interface Agents sollen in der Lage sein, die Last der Informationsflut zu verringern, indem sie im Interesse der Anwender handeln und ihnen bei Arbeitsabläufen intelligente Hilfe leisten Maes (1994).

Interface Agents sind Agenten, die sich mit dem Anwender die *Benutzerschnittstelle* als Arbeitsumgebung teilen. Sie können eine eigene Oberfläche mitbringen, oder gar personifiziert worden sein. Sie können aber auch, selbst unsichtbar, auf die Benutzerschnittstelle wirken und diese konfigurieren, modifizieren oder personalisieren, also auf ein bestimmtes Individuum und seine Bedürfnisse abstimmen.

Dieser Ansatz, der den Agenten erlaubt, aktiv zu werden und Arbeitsumgebung, Arbeitsablauf und Dialog mit dem Benutzer zu beeinflussen, steht im Konflikt mit dem Paradigma der direkten Manipulation, vergleiche hierzu auch den Abschnitt 3.2.3. Bei diesem Paradigma geht alle Aktion vom Benutzer aus. Programme reagieren, aber sie agieren nicht. Spätestens seit Microsofts Windows 98, bei dem erstmalig Interface-Agenten in Form von animierten, comichaften Charakteren mit ausgeliefert wurden und Windows erste Personalisierungsfunktionen anwendete, wie das Verbergen selten benutzter Menüeinträge, sind Interface-Agenten jedoch Teil der Ausstattung gängiger Benutzerschnittstellen.

Die ersten Versuche waren jedoch nicht bei allen Anwendern beliebt und wurden als

aufdringlich, beziehungsweise verwirrend, empfunden. Dies lag unter anderem an der Mißachtung grundlegender Wahrnehmungscharakteristiken des Menschen: Bewegungen im periphären Sehbereich sowie menschenähnliche Gesichtszüge ziehen unmittelbare Aufmerksamkeit auf sich. Diese Aufmerksamkeit ist wertvoll und sollte nur für einen angemessenen Gegenwert in Anspruch genommen werden, wie Eric Horvitz, der in die Entwicklung der animierten Agenten bei Microsoft involviert war, inzwischen in einem Interview in der New York Times (Markoff 2000) meint.

Wenn der Agent keine Möglichkeit hat zu beurteilen, ob die Person, an die er sich wendet, zur Zeit konzentriert ist, oder der Zeitpunkt zur Übermittlung einer Nachricht günstig ist, ist die Wahrscheinlichkeit, dass seine Aktivitäten störend wirken, sehr hoch. Deshalb wurden diese Funktionen in aktuelleren Versionen (zum Beispiel Windows XP, Vista) wieder zurückgenommen. Die Hilfe in aktuellen MS-Office-Versionen nutzen beispielsweise nicht mehr die personifizierten Objekte zur Anzeige von Hilfestellungen, sondern eine Sidebar. Die Zukunft sieht Microsoft in den Attentional User Interfaces (AUI), die die Aufmerksamkeit in angemessener Weise auf sich ziehen und dabei ein Benutzermodell und Bayesische Berechnungsmethoden anwenden, um gute Vorhersagen machen zu können.

3.3.2.1 Benutzungsschnittstellen

Die Benutzerschnittstelle eines Informationssystems, sowie die Schnittstelle des Arbeitsplatzes an dem ein Anwender das Informationssystem nutzt, sind die Umgebung, die sich der jeweilige Anwender und die Interface Agents teilen. Dabei kann ein Interface Agent im direkten Dialog mit dem Benutzer stehen. Dialoge können dabei sehr unterschiedlich ausfallen.

Wenn ein komplexer Arbeitsablauf automatisiert werden soll, beispielsweise ein Bestellvorgang, werden meist so genannte Assistenten (auch Wizards oder Druiden genannt) bevorzugt, die den Anwender durch eine Serie von aufeinander folgenden Formularen führen.

In Abbildung 3.4 ist als Beispiel ein Wizard aus der Bürosoftware OpenOffice abgebildet, der den Benutzer durch einen komplexen Prozeß führt.

Diese Form der Automatisierung, bei der der Anwender einerseits wichtige Entscheidungen selbst treffen kann, andererseits auch Schritte wieder rückgängig machen

kann und erkennt, wie weit der Vorgang schon abgeschlossen wurde, sind schon lange im Gebrauch und werden in grafischen User Interfaces häufig verwendet.

Zum Zweck der Hilfestellung in tutoriellen Situationen oder allgemein für kontext-sensitive Tipps werden häufig personalisierte Objekte (*Personas*) genutzt, die in direkten Dialog mit dem Anwender treten und ihm Fragen beantworten, Tipps anbieten oder ihm empfehlen, wie er seinen Arbeitsablauf optimieren kann. Der bekannte Assistent von MS-Office ist in Abbildung 3.1 zu sehen.

Eine etwas indirektere Form solcher Hilfestellung ist das Einblenden von Informationen in die vorhandene Oberfläche oder eine Sidebar. Das Ablenkungspotential ist bei animierten Charakteren, wie oben angesprochen, sehr hoch, da Menschen auf animierte Gesichter sehr stark mit Aufmerksamkeit reagieren.

Abbildung 3.4: Wizard von Open Office

Kontrolle Da Kontrolle aus psychologischen Aspekten sehr wichtig ist, ist es bedeutsam, das Gefühl der absoluten Kontrolle für den Anwender zu bewahren. Ansätze, bei denen die Agenten zu aktiv in die Interaktion des Anwenders eingreifen, sind in der Regel kritisch für das Autonomieempfinden des Anwenders, und damit ein Risiko für die Anwenderzufriedenheit. Die Diskussion in Abschnitt 3.2.3 weist auf die Probleme hin, die bei eingeschränkter Autonomie des Benutzers auftreten.

Die *Human Interface Guidelines* von Apple Computers führen das in der Fassung vom 16.6.2005, S. 42 so aus:

User Control

Allow the user, not the computer, to initiate and control actions. Some applications attempt to assist the user by offering only those alternatives deemed good for the user or by protecting the user from having to make detailed decisions. Because this approach puts the computer, not the user, in control, it is best confined to parts of the user interface aimed at novice users. Provide the level of user control that is appropriate for your audience.

Diese Betrachtungsweise geht von folgenden Thesen aus:

- Kontrolle durch den Benutzer hat Vorrang.
- „Gut gemeinte“ Automatisierung, die den Computer autonom handeln lässt, ist für manche Benutzer ungeeignet.
- Detaillierte Entscheidungen zu treffen, ist für unterschiedliche Benutzergruppen unterschiedlich wichtig.
- Für die richtige Gestaltung muss man seine Ziel-Benutzergruppe genau kennen.

Alternativ zur statischen Gestaltung und der Analyse der Benutzergruppe kommt in manchen Ansätzen auch eine adaptive Gestaltung in Frage, bei der die Bedürfnisse des Anwenders über einen gewissen Zeitraum hinweg beobachtet und das Benutzerprofil lernend erweitert wird.

Konfigurierbarkeit Die Konfiguration der autonomen Hilfsfunktionen und des Verhaltens der Agenten ist ein wichtiger Anwendungsfall. Zum einen ist die Akzeptanz und die Erwartungshaltung der Benutzer individuell sehr unterschiedlich, und sie kann sich auch nach Anwendungsfall und Fortschritt des Benutzers stark unterscheiden. Daher ist es für die Akzeptanz der Anwender entscheidend, ob und wie sie die Agenten konfigurieren können. Es ist auch ein wichtiger Mechanismus zur Kontrolle von autonomen Agenten, die ihre Aufgabe nach der Konfigurationsphase eigenständig ausführen.

Die Konfigurationsoptionen können einfache Entscheidungen enthalten, etwa ob und wie oft der Agent aktiv werden soll oder wie er seine Information meldet, beispielsweise als Popup-Meldung oder durch Dekoration eines Icons, welches den Agenten repräsentiert.

Die Konfiguration kann gegebenenfalls auch durch ein komplexes Werkzeug möglich sein. Ein Beispiel hierzu wäre das Workflow-Automatisierungswerkzeug „Automator“. Hier kann der Anwender Aktionen aus einer Bibliothek zu einer Pipeline zusammensetzen und jeden Verarbeitungsschritt mit gegebenen Parametern konfigurieren. Die so erstellten Arbeitsabläufe lassen sich manuell oder zeitgesteuert aufrufen – sie können auch an Ordner im Dateisystem gebunden werden, so dass dort gespeicherte Dokumente automatisch verarbeitet werden. Das Ziel ist es, durch Konfiguration solcher „Workflow-Robots“ eine umfangreiche, Automatisierung allen Benutzern zu ermöglichen (siehe Abbildung 3.5). Eine Automatisierung von Applikation ist ansonsten über die Programmierung in einer speziellen Skriptsprache möglich – eine Hürde für viele Anwender.

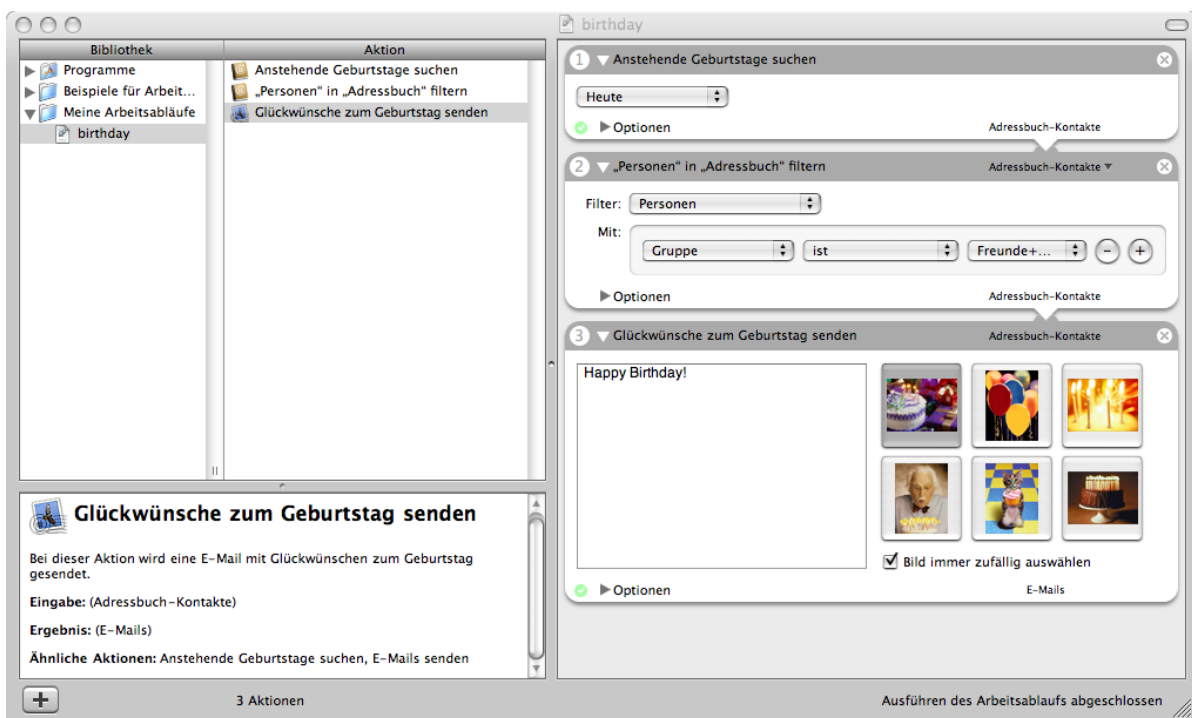


Abbildung 3.5: Automator: Automatisierte Arbeitsabläufe

Agenda-Transparenz Die Einsehbarkeit und Nachvollziehbarkeit der Handlungen und Aktionen der Agenten ist wichtig für die Vertrauensbildung und das Kontrollgefühl des Benutzers. Nur wenn er nachvollziehen kann, was der Agent tut, was er schon getan hat und was noch auf der Agenda steht, kann er entscheiden, ob der

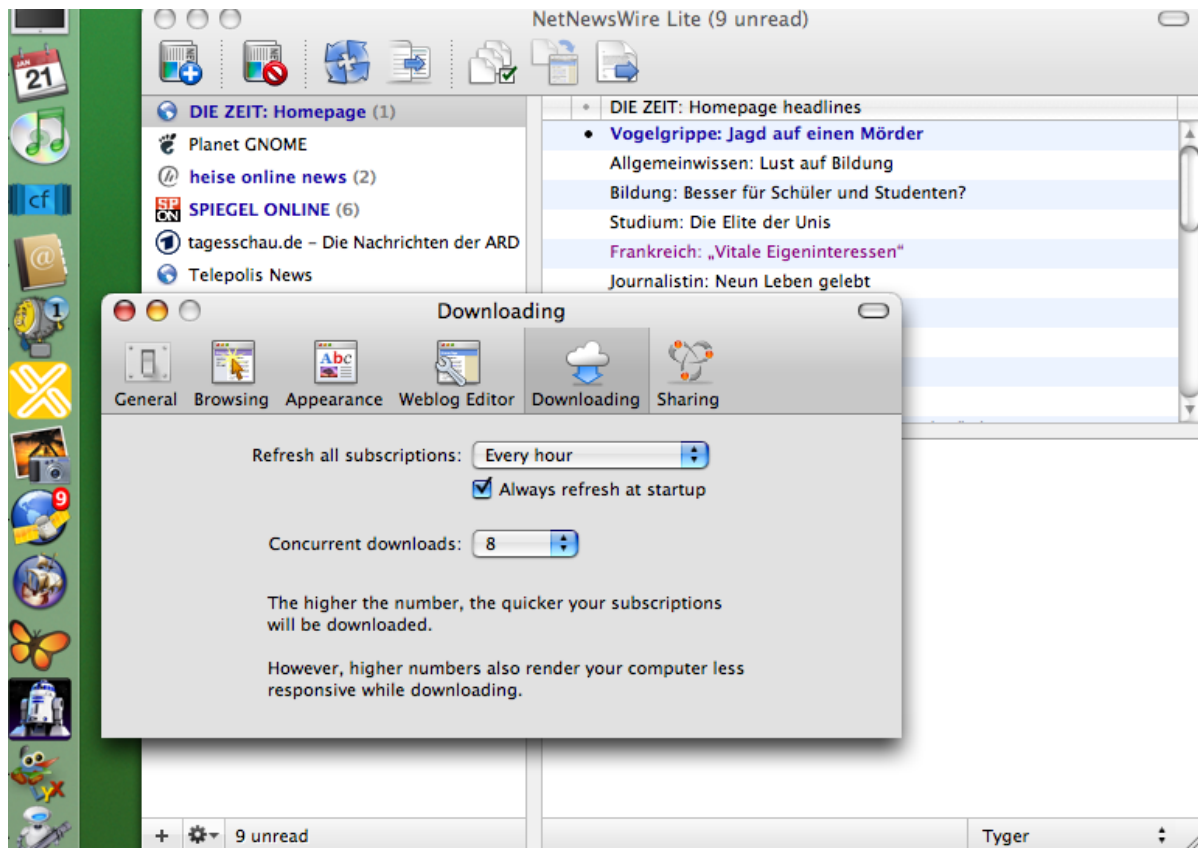


Abbildung 3.6: Ein Newsaggregator wird verwendet, um automatisch die neuesten Nachrichten aus verschiedenen Quellen zu sammeln.

Agent tatsächlich in seinem Sinne handelt. Er kann nur so entscheiden, ob der Agent unterbrochen werden sollte, und bekommt die Information, was noch zu tun bleibt.

Problembehandlung Bei allen Vorgängen können Probleme und Ausnahmen auftreten. Die Agenten brauchen Mittel, um solche Probleme zu signalisieren. Wenn ein solches Signal eine Reaktion des Anwenders erfordert, muss abgewogen werden, ob dies unmittelbar geschehen muss, oder ob der Anwender zu einem für ihn geeigneten Zeitpunkt eingreifen kann. Fehlermeldungen, die unmittelbar Nachrichtenfenster erzeugen und einen direkten Dialog mit dem Benutzer beginnen, sollten nur bei sehr ernststen Problemen herangezogen werden, etwa wenn Datenverlust droht. In vielen Fällen sind sicher Markierungen an Icons ein brauchbarer Hinweis auf Probleme.

Unaufdringlichkeit Als unaufdringlich gelten Annotationen außerhalb des direkten Fokus. Allerdings werden Annotationen im peripheren Sehbereich verstärkt wahrgenommen, wenn sie animiert sind. Unbewegte Annotationen sind somit den animierten vorzuziehen, solange es nicht dringend notwendig ist, die Aufmerksamkeit der Anwender zu erhalten.

Unaufdringlichkeit ist ein weiterer wichtiger Akzeptanz-Faktor. Als aufdringlich empfundene Agenten laufen Gefahr, schnell deaktiviert zu werden, bevor der Anwender gelernt hat, mit den Agentenfunktionen nutzbringend umzugehen (siehe hierzu auch die Grundlagen für Schnittstellen mit gemischter Initiative in Abschnitt 3.1.2).

Delegation Eine Teilaufgabe an einen Agenten zu delegieren ist eine attraktive Möglichkeit der Teilautomatisierung einer Aufgabe. Ein schon als Standard betrachtetes Beispiel ist die Aussortierung von Werbemüll durch intelligente Filter, die Werbung anhand erlernter Kriterien, Bayesischer Netze, unter Nutzung von benutzerdefinierter Regeln oder zentralisierten Blacklists, die bekannte Spammer enthalten. Das manuelle Aussortieren von Werbung kostet bei heutigem Werbeaufkommen im E-Mail-Verkehr zu viel Zeit und kann kaum noch geleistet werden (aktuelle Schätzungen gehen von einem Anteil von 70%–90% von unerwünschter Werbung am gesamten E-Mail-Verkehr aus, Tendenz steigend).

Wichtig ist für den Benutzer die Erkennbarkeit der Sortierkriterien und die Möglichkeit, bei falschen Entscheidungen korrigierend eingreifen zu können. Solche Korrekturen sollten auch nicht stupide wiederholt werden müssen, sondern sind sinnvoll.

lerweise langfristig auf das Regelwerk der Filter wirksam, so dass die Sortierqualität sich immer besser an die gewünschte Qualität annähert.

Aufgaben, die repetitiv ausgeführt werden müssen, sind gut delegierbar. Soll beispielsweise ein bestimmtes Thema in einer digitalen Bibliothek wiederholt recherchiert werden, um Neuerscheinungen zu beobachten, bietet es sich an, die Suchkriterien an einen Agenten zu übergeben und diesen in regelmäßigen Intervallen suchen zu lassen. Auch diese Form der Delegation ist mittlerweile eine gängige Grundfunktion von Informationssystemen. Beispiele finden sich etwa im Newsangebot von Google, in den meisten Job-Börsen oder in Form von Interpreten-Updates in Music-Stores. Da das Interesse an den so abonnierten Neuerungen zu den beobachteten Themen mit der Zeit abnimmt, ist es wichtig, dass das Abonnement mit der Zeit ausläuft, beziehungsweise der Anwender jederzeit die Delegation zurücknehmen kann.

Pfadsuche und -markierung Wenn Agenten bei der Informationssuche helfen, können sie im Informationsraum vorseilen und relevante Datenquellen ermitteln, Information auswählen, kategorisieren oder ihre jeweilige kontextbezogene Relevanz ermitteln. Sie können auch neue Zusammenhänge ermitteln oder Cluster berechnen. Ergebnisse dieser möglichen autonom ausgeführten Aufgaben müssen dem Anwender mitgeteilt werden, ohne ihn ständig zu unterbrechen. Möglichst kontextnah sollte die Information angeboten werden. Das bedeutet, falls es inhaltliche Bezüge zu bereits bekannten oder schon sichtbaren Informationen gibt, sollte das sichtbar werden. Diese Anforderungen lassen sich über *Pfadmarkierungen* erfüllen. Hierbei werden Marken an Icons oder Formularfelder, gegebenenfalls auch an Textstellen eingebettet, die bei Aktivierung zu den neuen Informationen leiten. Auch ein Newsfeed, bei dem neue Erkenntnisse der Agenten in einer Liste mit Nachrichten gesammelt werden ist eine mögliche Lösung, jedoch mit deutlich geringeren Kontextbindungen. Diese müssten in diesem Fall als Rückbezüge in der Nachricht enthalten sein.

Vertrauenswürdigkeit Da der Autonomiegrad, der dem Agenten mit gutem Gefühl vom Anwender zugestanden werden kann, stark vom Vertrauen in dessen Fähigkeiten und Zuverlässigkeit abhängt, muss der Agent besonders in der Phase der Vertrauensbildung genügend Feedback übermitteln und möglichst viel Einblick in seine Arbeitsabläufe geben können.

Für fortgeschrittene Benutzung, wenn bereits genügend Vertrauen aufgebaut wurde,

sollten diese Informationen jedoch versteckt werden können, um den Ablenkungsfaktor zu verringern. Auch ob Information über Arbeitsprozesse des Benutzers weitergeleitet wird, oder an zentraler Stelle gesammelt wird, ist ein wichtiger Vertrauensaspekt. Der Benutzer sollte in diesen Fällen über diesen Umstand informiert und um Erlaubnis gefragt werden. Falls Information über den Anwender an Dritte weitergeleitet wird, sollte die Funktion nicht ohne explizite Zustimmung des Benutzers eingeschaltet sein.

Relevanz Die Benutzerzufriedenheit hängt stark am Relevanzempfinden gegenüber den gelieferten Ergebnissen oder dem erzeugten Mehrwert. Um einen hohen Relevanzgrad zu erreichen, müssen neben geschickter Recherchemethodik auch stets Nutzerprofil und Informationskontext Beachtung finden. Dazu braucht der Agent Zugang zu einem differenzierten Benutzerprofil und zu den Kontextparametern. Darüberhinaus ist Zugang zu Informationssammlungen, die dem Benutzerinteresse zugeordnet werden können, etwa seine persönliche Informationsablage oder gemeinsam genutzte Sammlungen an Dokumenten, eine gute Vergleichsmöglichkeit zur Bestimmung der konkreten Relevanz neuer Informationen.

Lernen und Vergessen Agenten können lernen, zum Beispiel die angemessene Dialogform, das Benutzerprofil oder die Relevanzparameter können über die Zeit an die persönlichen Eigenschaften eines Benutzers angepasst werden. Da das Benutzerinteresse jedoch stets Änderungen unterworfen ist, ist es auch wichtig, nicht mehr benötigtes Wissen vergessen zu können.

Ordnung-Halten Agenten, die stets neue Information liefern oder stetig Annotationen in die Benutzungsschnittstelle einbetten, müssen auch dafür sorgen, Ordnung zu halten. Ansonsten machen sie dem Benutzer die zusätzliche Arbeit, seine Ordnung wiederherstellen zu müssen.

Als Beispiel sei an einen Informationsagenten gedacht, der ständig die Inhalte eines Ordners aktualisiert, anhand einer einmal definierten stehenden Anfrage. Die Menge der Inhalte des Ordners wird stets wachsen und der Agent wird stetig weitersuchen, wenn er nicht die Fähigkeit besitzt, die Ergebnisse nach Aktualität und Relevanz zu filtern.

Es wäre auch wünschenswert, dass der Agent anhand des Benutzerprofils abschätzen könnte, ob die generierte und von ihm gepflegte Ergebnismenge noch zu den Interessen des Nutzers passt. Gegebenenfalls müsste er auch die Entscheidung treffen können, dass der Nutzen seiner Recherche erschöpft ist, und mit dem Suchen aufhören. Natürlich sollte auch sein Auftraggeber, der Anwender, die Möglichkeit haben, die Arbeitsweise des Agenten durch direkten Dialog oder durch indirekte Interaktion, wie einem Relevance-Feedback-Mechanismus zu beeinflussen und zu justieren.

Aufdringlichkeit und Ablenkung Bei einem Informationssuchenden im *Anomalous State of Knowledge* (s.o.) ist durch die Unsicherheit, die dieser Zustand mit sich bringt, die Empfindlichkeit gegenüber Ablenkungen groß und damit die Möglichkeit gegeben, mit den gut gemeinten Hilfestellungen kontraproduktiv zu wirken.

Wie an anderen Stellen bereits diskutiert, ist die Unaufdringlichkeit der Agentenrückmeldungen wichtig. Es muss dafür gesorgt werden, das Ablenkungspotential zu minimieren und gleichzeitig den Wirkungsgrad der Recherche durch gut platzierte und kontextnahe Hinweise zu maximieren.

In dem Fall, dass sich jemand in eine Sackgasse recherchiert hat und Zusammenhänge oder weiterführende Suchterme nicht findet, die ihn in einer erfolgversprechenden Richtung suchen lassen würden, sind auch etwas sichtbarer platzierte Dialogelemente und weniger unaufdringlich gestaltete Hinweise eventuell produktivitätsfördernd. Das Problem für die Agenten besteht nun darin, solche Zustände zu erkennen.

3.3.2.2 Avatare, Personas

Avatare oder Personas sind personifizierte Repräsentanten für Agenten oder andere handelnde Personen, die bei einer verteilten Anwendung über das Netz mit dem lokalen Nutzer kommunizieren. Es sind demnach Stellvertreter, mit denen der Benutzer interagiert. Als solche können sie vor allem in stark dialogorientierten Szenarien angetroffen werden. Oft sind sie in Beratungssituationen oder in Tutor-Situationen zu finden.

Zu beachten ist bei dieser Gestaltungsform, dass Gesichtszüge sehr stark die Aufmerksamkeit anziehen. Oft ist gerade das aber eine Quelle der Irritation für den Anwender. Die Ergebnisse aus dem *Attentional- User-Interfaces*-Projekt (Horvitz u. a. 2003) belegen, wie begrenzt die Aufmerksamkeitsspanne des Menschen ist und wie

sorgsam mit dieser Ressource umgegangen werden muss. Die anthropomorphen Agenten von Microsoft Office, die ebenfalls von Horvitz entwickelt wurden, haben diesem Aspekt zu wenig Aufmerksamkeit geschenkt und wurden so von vielen Benutzern als störend empfunden.

Die Befürworter des „Persona-Effektes“ sehen jedoch das Potential, dass Personifizierung Ängste bei Benutzern abbauen kann und so dem Informationstransport zum Benutzer zugute kommen kann. In empirischen Studien ließen sich bislang jedoch keine eindeutigen positiven, allerdings auch keine eindeutig negativen Effekte feststellen. Die Benutzermeinung und die persönlichen Vorlieben für oder gegen Agenten unterscheidet sich individuell, aber die Präferenzen je Individuum sind klar ausgeprägt, das heißt es gibt Personen, die diese Gestaltungsform klar ablehnen, während andere Personen klar positiv reagieren. Jedoch gilt diese Beobachtung nur für Tests über einen beschränkten Zeitraum. Für Langzeittests liegen keine Erkenntnisse vor. (Dehn u. van Mulken 2000; van Mulken u. a. 1998)

3.3.2.3 Personalisierungsagenten

Bei komplexen Umgebungen mit vielen Gestaltungs- und Konfigurationsmerkmalen, wie es bei vielen Informationssystemen gegeben ist, ist die Aufgabe der Konfiguration nicht einfach. Sollen alle Einstellungen im Hinblick auf die persönlichen Vorlieben optimiert werden, müsste der Anwender seine Präferenzen kritisch hinterfragen und alle Optionen gut verstanden haben. Oft empfinden Benutzer das Ausführen dieser Aufgabe als Belastung, das Fehlen der Möglichkeit dazu jedoch als Einschränkung. Eine weitreichende passive Adaptierbarkeit ist also Voraussetzung für eine aktive Adaptivität. Benutzerprofil und hinreichende Parametrisierung sind vorauszusetzen, um den Agenten in die Lage zu versetzen, den Benutzer bei der auf ihn abgestimmten Anpassung seiner Arbeitsumgebung zu unterstützen.

Dieses Problem ist bereits aus der komplexen Arbeitsumgebung moderner Arbeitsplätze mit „Schreibtisch-Metapher“ bekannt. Um dem Benutzer die Aufgabe zu erleichtern, werden etwa in Microsoft Windows XP personalisierende Funktionen eingesetzt. Es wird zum Beispiel mitgezählt, wie oft Icons auf dem Schreibtisch tatsächlich benutzt werden, und nach einer gewissen Beobachtungszeit vorgeschlagen, die ungenutzten zu entfernen. So soll dem bekannten Desktop-Clutter-Problem entgegengewirkt werden, bei dem so mancher Anwender mit der Zeit die Übersicht über seine eigentlich als Erkennungsmerkmal gemeinten Ikonen verliert. Immerhin fragt

die Funktion hier nach, ob bestimmte Dinge entfernt werden sollen, und der Benutzer trifft eine bewusste Entscheidung. Andere Funktionen blenden ohne Rückfrage ungenutzte Menüpunkte aus, ohne den Benutzer diesbezüglich zu fragen, was häufig zu Verwirrung führt, da schon trainierte Bedienpfade plötzlich ganz anders aussehen oder erinnerte Menüpunkte plötzlich „fehlen“.

3.4 Entwurfsmuster

Nach dem großen Interesse, auf das Software-Agenten Mitte der 90er Jahre gestoßen sind, hat es auch zahlreiche Umsetzungen der Konzepte gegeben. Da es sich um ein Problemfeld handelt, bei dem sich bestimmte Lösungen für bestimmte Problemstellungen herausgebildet haben, werden im Folgenden die für Informationssysteme einschlägigen Entwurfsmuster behandelt und diskutiert, welche Kräfte im jeweiligen Zielkonflikt austariert werden.

3.4.1 Just In Time Information Retrieval

In der Dissertation von Rhodes u. Maes (2000) wird das Konzept des „Just In Time Information Retrieval Agents“ anhand von mehreren Beispielimplementationen erläutert. Bei diesem Konzept präsentieren Software-Agenten proaktiv Informationen, die jeweils zum lokalen Kontext eines Benutzers passen. Dieser Ansatz erhöht nach Rhodes' Erkenntnissen die Wahrscheinlichkeit, dass Informationen auf diesem Wege wahrgenommen werden. Der kognitive Aufwand, danach zu suchen, die Relevanz zu evaluieren und darauf zuzugreifen, wird verringert.

Teil der Arbeit sind auch Heuristiken zum Agentendesign, die dazu führen sollen, dass die Ergebnisse sichtbar, aber leicht zu ignorieren sind, um den Benutzer nicht zu stark von seiner Hauptaufgabe abzulenken.

Dieser Ansatz wird auch bereits im Abschnitt 3.3.1.3 beschrieben.

3.4.2 Eager Assistants

Eager Assistants lassen sich darauf trainieren, repetitive Aufgaben zu übernehmen und den Anwender so zu entlasten. Ein Beispiel ist EAGER (Cypher 1991), der mittels „Programming by Example“ seine Aufgaben lernt. EAGER nutzt als Interface

eine anthropomorphe Katze, sowie Einfärbung von Menüpunkten, für antizipatorisches Feedback. Das System lernt also vor allem im Dialog mit dem Anwender, parallel zu dessen eigentlichen Arbeitsabläufen. Es kann solche Assistants aber auch mit Skriptsprache oder grafischer Programmierschnittstelle geben – ein Beispiel hierfür wäre Apples „Automator“, wie in Abschnitt 3.3.2.1 erläutert.

Da die Skripte zwischen Benutzern ausgetauscht werden können, entsteht so auch ein überindividueller Mehrwert. Benutzer können ihre Lösungen für repetitive Aufgaben anderen zur Verfügung stellen, und so entsteht ein Lösungskatalog, der über die Zeit immer weiter verfeinert werden kann. Bei diesem Ansatz wird auf die verteilte Intelligenz der Anwendergemeinschaft zurückgegriffen. Das vereinfacht die Agentenarchitektur, vermeidet Probleme mit unzureichender künstlicher Intelligenz und entlastet den Anwender, der den Agenten nun nicht mehr speziell trainieren muss.

Eager Assistants haben Funktionen, um Anwendungen auf die gleiche Weise zu steuern, wie der Benutzer dies tut. Ihre Funktionen bestehen somit aus Ketten von kleineren Teilaufgaben, die Benutzer ohne die Assistenten manuell ausführen müssten.

Da Eager Assistants im Auftrag des Benutzers handeln und eher langweilige, leicht zu delegierende Teilaufgaben übernehmen, ist das Konfliktpotential gegenüber den Anwender gering. Dieser kann sich nun besser auf seine Hauptaufgaben konzentrieren und erreicht diese in weniger Arbeitsschritten auf höherem Abstraktionsniveau.

Kognitive Last kann jedoch anfallen, da der Benutzer für die Workflow-Planung die Möglichkeiten der Eager Assistants verstehen und berücksichtigen muss. Er muss bei dem Ansatz der von einer Community erstellten Skripte auch gegebenenfalls nach Teillösungen recherchieren, die zu seinem Problem passen. Somit verlagert sich im Kontext der Informationssysteme ein Teil der kognitiven Last auf die Auswahl und Definition der Arbeitsschritte für die Software-Assistenten. Da diese Arbeit jedoch bei regelmäßig wiederholten Workflows nur anfangs anfällt, lohnt sich die investierte Mühe auf Dauer. Bei Workflows, die nur einmalig oder selten auftauchen, oder bei Informationssystemen, die sich nicht an wiederkehrende Benutzer sondern an gelegentliche Nutzer wenden, etwa in öffentlichen Terminals in Bibliotheken, lohnt sich der Aufwand weniger.

3.4.3 Wizards

Wizards helfen, komplexe Workflows hinter einer vereinfachten Fassade zu verbergen. Der Benutzer wird durch Arbeitsschritte auf einer höheren Abstraktionsebene geleitet und muss nur die notwendigen Entscheidungen treffen. Er kann auch Entscheidungen rückgängig machen und behält so die Kontrolle. Er muss sich nicht ängstigen, dass eine Ad-Hoc getroffene Entscheidung nicht umkehrbar ist und kann zunächst sehen, wohin seine Entscheidung führt.

Solche Wizards haben sich zum Beispiel für Installationsroutinen und erstmalige Konfiguration von Software bewährt. Auch zum Erstellen neuer Objekte, etwa in Textverarbeitungen oder in integrierten Softwareentwicklungsumgebungen, sind Wizards häufig verwendete Entwurfsmuster.

Der Nachteil beim Einsatz von Wizards liegt ebenfalls in der Abstraktion. Falls etwas nicht erwartungsgemäß funktioniert, etwa die Installation einer Software fehlschlägt, ist es oft schwer zu erkennen, wo der Fehler lag. Der Benutzer wird dann doch oft mit Protokollen konfrontiert, die eine detaillierte, systemnahe Sicht wiedergeben und den Vorteil der Wizards zunichte machen. Deshalb ist es bei Wizards besonders wichtig, bei der Gestaltung insbesondere Wert auf Ausnahme- und Fehlerbehandlung, beziehungsweise Fehlervermeidung zu legen.

3.4.4 Status-Manager

Eine spezielle Aufgabe bei modalen Arbeitsabläufen liegt in der Verwaltung des Status oder sogar mehrerer Statusmerkmale.

Ein Statusmerkmal kann etwa der Verbindungsstatus zu einem Dienst oder Server sein. Wenn beispielsweise in einem Informationssystem länger dauernde oder wiederkehrende Recherchen an Agenten delegiert werden können, ist es für den Anwender wichtig, den Status und den Fortschritt bei Bedarf erkennen zu können.

Bei Status-Managern ist es die Hauptaufgabe, den Status zu verwalten und den Anwender in dieser Hinsicht zu entlasten. Bei Ausnahmesituationen muss der Status-Manager aber auch den direkten Dialog ermöglichen und auf das Problem aufmerksam machen.

3.4.5 Aktive Hilfe

Das Konzept der Aktiven Hilfe verspricht, den Kontext zu verfolgen und Tätigkeiten des Anwenders mit den möglichen Zielen zu vergleichen.

Dabei wird versucht festzustellen, ob der Anwender zu weit von seinem optimalen Handlungspfad abweicht. Wenn es zu solchen Abweichungen kommt, wird dem Benutzer ein Vorschlag gemacht, wie er sein Handeln optimieren kann, oder es werden hilfreiche Informationen eingeblendet.

Insbesondere kann eine aktive Hilfe versuchen, den Anwender aus Sackgassen zu befreien, etwa wenn dieser mehrmals dieselbe Suchanfrage absendet, die keine oder zu wenige Ergebnisse liefert.

Ein Problem ist hierbei die Modellierung der optimalen Pfade für einen Workflow. Wenn das Modell nicht ausreichend funktioniert, ist es nicht einfach, dem Benutzer ausreichende Hilfestellung passend zum Kontext anzubieten.

Es gibt Ansätze, das Modell sukzessive mit maschinellem Lernen zu verbessern. Am Ende der Lernphase ist das Angebot der aktiven Hilfe dann potentiell wesentlich besser als zu Anfang. Das Bedürfnis nach optimaler Hilfe ist für Benutzer allerdings gerade zu Anfang gegeben. Es ist also notwendig schon mit angemessener Ausgangsbasis zu starten. Die Hilfe kann dann versuchen, sich durch Lernen auf individuelle Unterschiede einzustellen.

3.4.6 Path Marker

Die Path-Marker-Agenten treten nicht in direkten Dialog mit dem Anwender, sondern agieren wie eine Vorhut auf der Suche nach Lösungswegen und Problemlösungen, die zum Kontext des Anwenders passen. Insofern sind sie verwandt mit den Aktive-Hilfe-Agenten. Sie markieren allerdings Informationselemente mit Hinweisen, so dass Benutzer sich anhand dieser Markierungen ihren Pfad suchen können. So kann ein Pfad beispielsweise von einer recht allgemeinen Suche ausgehen und ein Path-Marker-Agent kann durch geschickte Extraktion von Entitäten, wie Autoren oder einschlägigen Termen, helfen, die große Ergebnismenge geschickt einzugrenzen.

In modernen Entwicklungsumgebungen, wie beispielsweise der integrierten Entwicklungsumgebung Eclipse, spielen Pfadmarkierungen eine wichtige Rolle. Dort werden ständig im Hintergrund die Eingaben der Entwickler mit Markierungen versehen, die auf Fehler, Konventionsverletzungen, mögliche Codevervollständigungen oder Optimierungsmöglichkeiten hinweisen. Da die Hinweise oft sehr wertvoll und meist

zutreffend sind, wird diese intelligente Unterstützung gerne akzeptiert und erlaubt die Konzentration auf die wesentlichen Code-Eigenschaften.

3.5 Strategische Unterstützung durch Agenten in digitalen Bibliotheken

In digitalen Bibliotheken und ähnlichen Informationssystemen zur wissenschaftlichen Informationsversorgung gilt es oft, ein übergeordnetes strategisches Ziel zu erreichen. Beispielsweise möchte man zu einem Fachthema die einschlägige Literatur der letzten zwei Jahre zusammenstellen, eine Reihe von Experten ausfindig machen oder im Hinblick auf eine wissenschaftliche Fragestellung den Stand der Technik abklären.

Die Vielzahl an zu berücksichtigenden Quellen und der durch Zugangsbeschränkungen erschwerte Zugriff auf die Information sowie das generelle Recherche- und Qualitätsbewertungsproblem bei der wissenschaftlichen Informationsversorgung erfordern ein hohes Maß an strategischem und taktischem Können. Der Suchende muss über das Fachgebiet und die einschlägigen und qualitativ hochwertigen Informationskanäle wie Zeitschriften oder Tagungen und deren Reputation und Zuverlässigkeit gut Bescheid wissen.

Ein wichtiger Anwendungsfall ist das Auffinden und Verfolgen von Verknüpfungen zu weiterführender Literatur. Dies kann zitierte oder zitierende Literatur sein oder auch Literatur, die, ausgehend von einschlägigen Dokumenten, über die Autoren dieser Werke und deren Kollaborationsumfeld, ausfindig gemacht wird.

Auch die Tiefensuche in den Journalen, in denen einschlägige Treffer vorkamen, oder in den Tagungsbänden, in denen zum aktuellen Problem einschlägige Literatur veröffentlicht wurde, ist ein gängiges Muster des Suchverhaltens, das häufig zum Erfolg führt. Bates (1990) nennt diese Form der Tiefensuche im Rahmen einer Strategie ein *Stratagem*.

Einige Stratageme sind relativ häufig zu beobachten, da sie erfolgversprechend sind und sich als „Best Practice“ etabliert haben. Bates hat Stratageme beschrieben und benannt. Einige exemplarische Stratageme sind:

- *Journal Run*

Beim Journal Run wird nach einigen einschlägigen Treffern in einem Journal ein

oder mehrere Jahrgänge desselben Journals nach weiteren Veröffentlichungen zu demselben Thema durchsucht.

- *Conference Run*

Beim Conference Run wird analog zum Journal Run nach einschlägigen Treffern in einem Proceedings-Band die ganze Serie gezielt nach weiteren Treffern durchsucht.

- *Homepage Run*

Bei mehreren einschlägigen Treffern eines bestimmten Autors wird die Veröffentlichungsliste auf dessen Homepage nach weiteren Treffern durchsucht.

- *Citation Run*

Ausgehend von einem einschlägigen Treffer werden gezielt die Artikel verfolgt, die diesen zitieren oder von diesem zitiert werden. Dies kann iterativ in beliebiger Tiefe geschehen.

Um solche Taktiken, Strategeme und Strategien erfolgreich entwickeln und anwenden zu können, benötigen die Anwender eines Informationssystems ein nennenswertes Maß an *Informationskompetenz* (siehe Abschnitt 3.5).

Der Systemunterstützung werden von Bates vier Ebenen zugeordnet, zuzüglich der Möglichkeit, eine Basishandlung bei der Informationssuche ohne Computerunterstützung zu leisten, siehe hierzu Tabelle 3.2.

Bates gibt zudem Empfehlungen, auf welche Ansätze der Systemunterstützung in Bezug auf die Abstraktionsebenen sich die Forschung und Entwicklung zunächst konzentrieren sollte: Während die Systeme sich zu der Zeit zumeist darauf beschränkten, Basisaktionen anzuzeigen und auswählbar zu machen, sollte sich die Forschung auf das Anbieten von taktischen und strategischen Funktionen konzentrieren und diese gegebenenfalls in Bezug auf den Kontext und das Verhalten des Benutzers vorschlagen. Wenn dieses Zwischenziel erreicht ist, kann über das Anbieten komplexer Strategien geforscht werden. Experimentelle Information-Retrieval-Systeme können den Bereich der komplexen Automatisierung ohne notwendige Benutzerkontrolle erschließen. Die daraus resultierende Matrix findet sich in der Abbildung 3.7 (nach Klas (2007, S. 47)).

Der Komplex der strategischen Unterstützung wird im Zusammenhang mit der Projektbeschreibung DAFFODIL im Abschnitt 4.2 wieder aufgegriffen.

3 Arbeitsteilung in agentengestützten Informationssystemen

Ebene	Name	Eigenschaft
0	Kein System	Reine Benutzerebene; keine Vorschläge/ Aktionen vom System
1	Möglichkeiten auflisten	Vorschläge von Möglichkeiten auf Anfrage; keine Durchführung
2	Aktionen auf Kommando	Kann Aktionen ausführen (passiv!); kein Analysieren der Suche
3	Überwachen, Vorschläge	Überwachen, Analysieren einer Suche, Vorschläge-Machen - auf Benutzeranfrage - bei Entdecken
4	Automatische Ausführung	Automatisches Ausführen einer Aktion - Information an den Benutzer - Keine Information

Tabelle 3.2: Ebenen der Systemunterstützung (Bates 1990)

Systemaktivität System- unterstützung	Basis Aktion	Taktik	Strategem	Strategie
Keine Systemunterstützung	Suchaktivität vom Benutzer ausgeführt			
Anzeige möglicher Aktivitäten	Heutige IR Systeme	Gebiet der Forschung		Später zu Unter- suchen
Manuelles Ausführen von Aktionen				
Beobachten und Vorschlagen	Nicht Unter- suchen			
Ausführen von Aktionen				
	Experimentelle IR Systeme			

Abbildung 3.7: Matrix zu Systemunterstützung in Bezug auf Aktivitätsebenen.

Exkurs: Informationskompetenz

Informationskompetenz bedeutet nach Kuhlen, sich den Zugriff auf Information beschaffen, diese einschätzen und anschließend aktiv nutzen zu können, und zählt zu

den Basiskompetenzen der Informationsgesellschaft. (Knorz u. Kuhlen 2000)

Informationskompetenz ist Voraussetzung für informationelle Autonomie, die Fähigkeit, mündig und selbstbestimmt mit Wissen und Information umzugehen.

Hierzu schreibt Kuhlen (1999, S. 145f.) allerdings kritisch:

Trotz aller Möglichkeiten ist gegenwärtig keineswegs ein Paradies informationeller Autonomie auszumachen. Dafür sorgt schon ein im Informationsgebiet auch bei anderen Gelegenheiten immer wieder auftretendes Informationsparadoxon: Informationsmärkte, dafür konzipiert, den Umgang mit Information immer leichter zu machen, in den politischen Verlautbarungen dazu bestimmt, das Ziel informationeller Autonomie für aufgeklärte Bürger in der Informationsgesellschaft zu befördern, scheinen Erarbeitung und Nutzung von Information immer aufwendiger und schwieriger zu machen.

Eine erfolgreiche Recherche in einem komplexen wissenschaftlichen Themengebiet erfordert angemessene strategische Fähigkeiten. Der Suchende muss seine Handlungen sorgsam planen und ist dennoch oft in Situationen, in denen eine Orientierung schwer fällt. Dazu kommt, dass die unübersichtliche Informationslandschaft die Orientierung und Handlungsfähigkeit einschränkt.

Die vermeintliche informationelle Kompetenz, die viele Anwender derzeit durch Suchmaschinen wie Google, MSN-Search und Yahoo! zu haben glauben, ist trügerisch, insbesondere durch das schon genannte Invisible-Web-Problem. Es entstehen oft nicht wahrgenommene Erschließungslücken durch den nur eingeschränkt vorhandenen Durchdringungsgrad der Suchmaschinen in Bezug auf strukturierte wissenschaftlich relevante Angebote.

Selbst der auf wissenschaftliche Veröffentlichungen spezialisierte Dienst von Google, namens Scholar-Google⁵, weist deutliche Defizite in Abdeckung und Präzision auf. Eine Studie des Informationszentrums Sozialwissenschaften, Mayr u. Walter (2006), untersuchte den Abdeckungsgrad für Zeitschriftenartikel, da die genauen Quellen bei Google-Scholar undokumentiert sind. Die Studie zeigt Defizite in der Abdeckung und Aktualität des Google-Scholar-Index. Weiterhin macht die Studie deutlich, welche Webserver die wichtigsten Datenlieferanten für den neuen Suchdienst sind und welche wissenschaftlichen Informationsquellen im Index unterrepräsentiert sind:

⁵<http://scholar.google.com>

Die Pluspunkte von Google Scholar liegen in seiner Einfachheit, seiner Performanz und letztendlich seiner Kostenfreiheit. Die Recherche in Fachdatenbanken kann Google Scholar trotz sichtbarer Potenziale (z.B. Zitationsanalyse) aber heute aufgrund einer Reihe von Schwächen nicht ersetzen. (ebenda)

3.5.1 Unterstützung durch hochwertige Suchfunktion

Durch die Bildung hochwertiger Suchfunktionen kann die Suchkompetenz der Anwender gesteigert werden. Hochwertige Suchfunktionen sind hierbei so zu verstehen: Eine Funktion zur Suche nach Information stößt einen komplexen Prozeß an, der eine Reihe von einzelnen Aktionen und Anfragen so verbindet, dass eine Tiefen- oder Breitensuche zu einer Funktion zusammengeführt wird. Für den Benutzer sollen so wertvolle Bausteine für eine Suchstrategie oder Suchtaktik angeboten werden.

Ein Suchindex, der Inhalte verschiedener Quellen zusammenfasst, bietet ebenso eine hochwertige Suchfunktion wie eine föderierte Suche, die Anfragen zum Suchzeitpunkt an diverse heterogene Suchdienste verteilt. Ebenso ist eine Suchfunktion hochwertig, die iterativ oder rekursiv über einer breiten Menge von Daten neue Relationen zwischen Dokumenten oder Metadaten ermittelt.

3.5.2 Föderierte Suche

Bei einer föderierten Suche wird die Suchanfrage nach einer Transformation für verschiedene Suchdienste aufbereitet, verteilt und die Ergebnisse aggregiert und gemeinsam präsentiert. Eine föderierte Suche ist im Vergleich zu einem integrierten Suchindex in Bezug auf die Laufzeit im Nachteil, da die Suchanfrage und die Ergebnisse zum Anfragezeitpunkt prozessiert werden müssen. Dafür lassen sich auf diesem Wege auch Systeme anbinden, deren Inhalt nur über eine Suchfunktion erreichbar ist. Auf diesem Weg läßt sich das „Deep Web Problem“ zu einem Teil mindern.

3.5.3 Integrierte Suche

Bei der integrierten Suche werden Daten zusammengeführt und mittels Suchmaschine oder mittels intellektueller Verschlagwortung inhaltlich hochwertig erschlossen.

Der so erstellte Suchindex steht anschließend in sehr effizienter Form zur Verfügung und alle indexierten Inhalte können mit einer homogenen Anfragesprache erschlossen werden.

Eine Heterogenitätsbehandlung bezieht sich in diesem Fall nur noch auf semantische Unterschiede im Vokabular oder Metadatenschema. Für den Benutzer heißt das im Idealfall, dass er sich mit der ursprünglichen Herkunft der Daten nicht zu befassen braucht – aus seiner Sicht befragt er eine einzige hochwertige Datenquelle.

3.5.4 Schalenmodell

Beim Schalenmodell (Krause 2006) wird davon ausgegangen, dass es in Zeiten der Deregulierung und der weltweiten Vernetzung niemals möglich (und auch nicht sinnvoll) ist, alle für einen Bereich relevanten Datenquellen in hoher Qualität in einer integrierten Suche zusammenzuführen. Dabei wird von einem integrierten, hochwertig inhaltlich erschlossenen Kern ausgegangen, der eine kritische Menge von hochwertigen Daten in einer integrierten Suche zusammenführt. Auf den Eigenschaften dieser Daten können auch deduktiv Erkenntnisse über weiterführende Eigenschaften der Daten ermittelt werden, wie etwa statistische Verteilung der Schlagworte oder Zusammenhänge zwischen Autoren, um etwa Kollaborationsnetze zu berechnen.

Neben diesem integrierten Kern werden weitere Datenquellen mit niedrigerer Erschließungsqualität, etwa via föderierter Suche, an das Gesamtsystem angeschlossen. Hierbei helfen die vorhandenen Daten des Kernbereiches, den gesamten Qualitätsstandard zu heben. So lässt sich ein größerer Informationsraum aufspannen, der „blinde Flecken“ in der Inhaltserschließung schrumpfen lässt.

Genutzt wird das Konzept des Schalenmodells beispielsweise im Informationsportal Vascoda⁶, wo ein auf Crosskonkordanzen basierender Heterogenitätsservice, der als Term-Umschlüsselungs-Dienst fungiert, zum semantischen verbinden der “Schalen” eingesetzt wird (Mayr u. Walter 2007).

Hindernisse zur Umsetzung können weiterhin kommerzielle Hürden und Lizenzfragen sein. Auch Heterogenitätsprobleme lassen sich in einem solchen Szenario nur auf ein erträgliches Maß reduzieren, aber kaum ganz verhindern. Somit muss sich der Nutzer eines solchen Systems auch zum Teil noch mit der komplexen Struktur des Systems auseinandersetzen, um die Effekte richtig einschätzen zu können.

⁶<http://www.vascoda.de/>

Das Modell des "Semantic Web", welches versucht, über abgestimmte Ontologien die semantischen Lücken zwischen den diversen Informationslieferanten im Internet zu schließen, steht hingegen ebenfalls vor dem Problem der notwendigen semantischen Homogenisierung oder notfalls Abbildung, wenn die Grenzen einer geschlossenen Informationswelt übertreten werden. Abstraktere Ontologien, welche Konzepte allgemeiner definieren, helfen nur eingeschränkt weiter. Daher werden auch bei diesem Ansatz semantische Transfermechanismen benötigt, die die Ontologien und deren Konzepte an den Schnittstellen so koppeln, dass eine globale Semantik beschreibbar wird. (Krause 2008) kommt zu dem Schluß, dass die beiden Ansätze sich somit gut ergänzen und nicht im Widerspruch stehen.

3.5.5 Informationeller Mehrwert

Durch die Integration von Datenquellen zu Informationssystemen mit breitem inhaltlichen Spektrum entstehen auch neue Chancen auf informationellen Mehrwert. Durch die Zusammenführung von vormals unverbunden und verteilt gespeicherten Fakten und Metadaten können neue Funktionen und Analysemethoden umgesetzt werden, die auf den ursprünglichen Daten in dieser Form nicht möglich gewesen wären. Insbesondere Breitenanalysen, wie die Berechnung von Koautoren-Netzwerken, oder Tiefenanalysen, wie die Verfolgung von Inter-Text Relationen (wie bei Zitationsrelationen), profitieren von der breiteren und tieferen Datenbasis.

Im folgenden Kapitel soll das Projekt DAFFODIL vorgestellt werden, das auf Grund seiner Zielsetzung die oben genannten Ansätze verfolgt, um eine strategische Unterstützung der Anwender möglich zu machen. Dieses Ziel wird unter Anwendung des Agentenkonzeptes und softwareergonomischer Gestaltungsrichtlinien verfolgt.

4 Daffodil

DAFFODIL ist ein verteiltes Informationssystem, das qualitativ hochwertige Angebote aus dem Bereich der Informatik nach dem Prinzip des Schalenmodells verknüpft (siehe Abschnitt 3.5.4) – darunter die bekannte ACM Digital Library oder die Veröffentlichungen des Springer Verlages. Die verteilten Informationen werden durch föderierte Suche zusammengeführt und durch Homogenisierung, Anreicherung und Analyse zu einem System von Suchdiensten und einem integrierten Arbeitsplatz zusammengeführt.

Die ergonomische Gestaltung von DAFFODIL basiert auf dem WOB-Modell. Das Modell bildet einen Katalog von Gestaltungsrichtlinien auf einer pragmatischen Abstraktionsebene zwischen den konkreten Handlungsanweisungen der Style-Guides für die Softwareentwicklung einerseits und den abstrakten Vorgaben der rechtsverbindlichen Normen andererseits. Das WOB-Modell wird in Abschnitt 4.3 beschrieben.

4.1 Ausgangspunkt und Zielsetzung

Ausgangspunkt für das Projekt DAFFODIL war die Beobachtung der Informationswissenschaften, dass eine intelligente Unterstützung des Benutzers bedeuten muss, einen großen Teil der Problemlösungsintelligenz dem Benutzer zu überlassen und ihn kontextspezifisch zu unterstützen. Die Kernthesen zu einer adäquaten Vorgehensweise wurden von Bates (1990) formuliert:

Auf dem Weg zur Erreichung des strategischen Ziels, also des Findens aller zur Lösung des aktuellen Problems notwendigen Information, werden unterschiedliche *Taktiken* angewendet. Die Taktiken und einzelnen Schritte (*Moves*) ergänzen sich zu komplexeren Handlungsabläufen, um etwa eine Datenquelle systematisch zu durchforschen und alle relevanten Treffer – beispielsweise aus einem einschlägigen Journal – zu finden. Diese komplexeren Handlungen nennt Bates *Stratageme*.

Da es also bei der Recherche komplexe Handlungen und ein allgemeines Verständnis der Taktiken, Informationsquellen und der strategischen Möglichkeiten notwendig ist, wird klar, dass eine ganze Menge Informationskompetenz für ein zufriedenstellendes Ergebnis notwendig ist. Hinzu kommt die Notwendigkeit, die eigenen Ziele zu erkennen und auf dem Weg zum Ziel sein Informationsbedürfnis zunehmend treffender beschreiben zu können.

Bates charakterisiert die Qualität von Informationssystemen, indem die systemseitige Unterstützung für Handlungen auf den Ebenen der Basishandlungen, der Taktiken und der Strategeme sowie der Strategie beurteilt wird. Zur Diskussion der strategischen Unterstützung siehe auch Abschnitt 3.5.

Ausgehend von dieser Betrachtungsweise wurde mit dem Projekt DAFFODIL eine Infrastruktur für eine integrierte Suche in heterogenen digitalen Bibliotheken eines Fachgebiets unter Zusammenführung der Ergebnisse erstellt. Darüber hinaus erfahren die Benutzer bei ihrer Suche strategische Unterstützung durch die angebotenen höheren Suchfunktionen, umgesetzt in einer Reihe von integrierten Werkzeugen (Fuhr u. a. 2000).

4.2 Projektbeschreibung

DAFFODIL verbindet in seiner Oberfläche in natürlicher Weise Browsing- und Suchstrategien über eine breite Auswahl von Informationsquellen. Grundsätzlich sind die Benutzer dabei frei in der Wahl ihrer individuellen Suchstrategie, werden jedoch bestmöglich strategisch unterstützt. Zudem erfährt der Benutzer Unterstützung bei der Durchführung sowohl von beliebten Suchtaktiken oder -strategemen als auch bei der Kombination solcher, die unter anderem zu umfangreicheren Suchplänen führen können. Bates hat zahlreiche auf empirische Beobachtung gestützte, erfolgreiche Taktiken für die Informationssuche identifiziert Bates (1987, 1989), welche zu weiten Teilen in DAFFODIL angeboten werden. Im Folgenden wird auf die grafische Oberfläche einschließlich der wesentlichen Werkzeuge eingegangen. Eine kurze Darstellung der Ergebnisse einer Evaluation der strategischen Unterstützung der Benutzer und der angebotenen proaktiven Funktionen folgt im Anschluss daran. Zur differenzierten Diskussion zu Suchtaktiken findet sich eine Diskussion bei Klas (2007).

4.2.1 Projektverlauf

Das Projekt DAFFODIL wurde von der deutschen Forschungsgemeinschaft (DFG) im Schwerpunktprogramm "Verteilte Verarbeitung und Vermittlung digitaler Dokumente"(V3D2) gefördert.

Die Projektlaufzeit von DAFFODIL teilt sich in zwei Projektabschnitte:

2000–2002 Die erste Projektphase war eine Kooperation der Universität Dortmund mit dem Informationszentrum Sozialwissenschaften in Bonn. Ziel der Projektphase war die Konzeption und Implementation einer verteilten agentenbasierten Infrastruktur zur föderierten Suche in heterogenen digitalen Bibliotheken. Die Infrastruktur sollte eine ergonomisch optimale Benutzungsoberfläche im Sinne des WOB-Modells erhalten, die ermöglicht, höhere Suchfunktionen über der Dienstestruktur zu nutzen.

2002–2005 Die zweite Projektphase sollte die Möglichkeiten des Anwenders hin zu einer ganzheitlichen Rechercheumgebung erweitern, die auch Unterstützung für die Arbeit nach und neben der Recherche bietet. Daneben sollte auf eine möglichst umfassende strategische Unterstützung des Anwenders hingearbeitet werden.

Im Frühjahr 2005 wurde die zweite Projektphase von DAFFODIL erfolgreich abgeschlossen.

4.2.2 Modellierung der Rechercheoperationen

Während herkömmliche Suchsysteme oft nur einfache, grundlegende Suchoperationen unterstützen, zielt DAFFODIL darauf ab, durch die enge Integration von höheren Suchfunktionen gerade Nichtexperten eine effektivere Suche zu ermöglichen. Bates führte in diesem Zusammenhang drei Abstraktionsebenen für die Kategorisierung von Suchfunktionen ein:

- *Moves* sind einfache Handlungen wie die Eingabe von Termen in ein Suchformular,
- *Taktiken* sind als eine oder mehrere kombiniert eingesetzte grundlegende Suchoperationen zu verstehen: etwa die Schritte, die nötig sind, um eine Suchanfrage zu verfeinern, oder die Generalisierung eines Suchbegriffs.

- *Stratageme* stellen eine komplexe Abfolge von Aktionen aus Grundoperationen und Taktiken dar, die die Strukturen einer Domäne zur Suche ausnutzen.
- Unter einer *Strategie* soll dann ein vollständiger Plan zur Ausführung einer Recherche verstanden werden. Strategien setzen sich im Allgemeinen aus einer Reihe von Grundoperationen, Taktiken und Stratagemen zusammen.

DAFFODIL setzt auf der Ebene der Stratageme an und bietet durch strategische Unterstützung den Benutzern die Möglichkeit zur Verfolgung komplexer Suchstrategien. Ein Stratagem soll in Form eines in den Recherchedesktop eingebetteten Werkzeugs zur Verfügung stehen. Die so erzielten Zwischenergebnisse in Form von Dokumenten, Metadaten, Personen oder Suchtermen lassen sich nutzen, um weitere Taktiken und Stratageme anzustoßen. Dieses iterative Vorgehen entspricht dem Berrypicking-Verhaltensmodell von Bates.

Zusätzlich werden die Arbeitsabläufe, die von den Stratagemen in DAFFODIL unterstützt werden, in fünf verschiedenen Phasen des “Digital Library Life Cycle” nach Weibel u. Miller (1997) und Paepcke (1996) unterteilt: *Discover*, *Retrieve*, *Collate*, *Interpret* und *Re-Present*. Dieser Zyklus ist in Abbildung 4.1 dargestellt.

Einerseits soll so die Komplexität des Systems verringert werden – und andererseits die Benutzer in jeder dieser Phasen gezielt unterstützt werden können:

- *Discover*:
In dieser Phase steht die Suche und Auswahl der Datenquellen durch den Benutzer im Vordergrund.
- *Retrieve* :
In dieser Phase durchsucht der Benutzer die verschiedenen vom ihm ausgewählten Datenquellen.
- *Collate*:
In dieser Phase erfolgt die strukturierte Ablage der gefundenen Informationen.
- *Interpret*:
In dieser Phase nimmt der Benutzer die Interpretation der von ihm oder von Gruppenmitgliedern abgelegten Informationen vor.
- *Re-Present*:
In dieser abschließenden Phase erfolgt die Darstellung des neu gewonnenen Wissen z. B. in Form einer wissenschaftlichen Publikation.

Die Umsetzung dieser fünf Phasen bietet den Benutzern ein in sich geschlossenes Serviceangebot. Strategische Unterstützung im Suchprozess erfährt der Benutzer durch eine Vielzahl von integrierten Werkzeugen, die sich durch höhere Suchfunktionen auszeichnen.

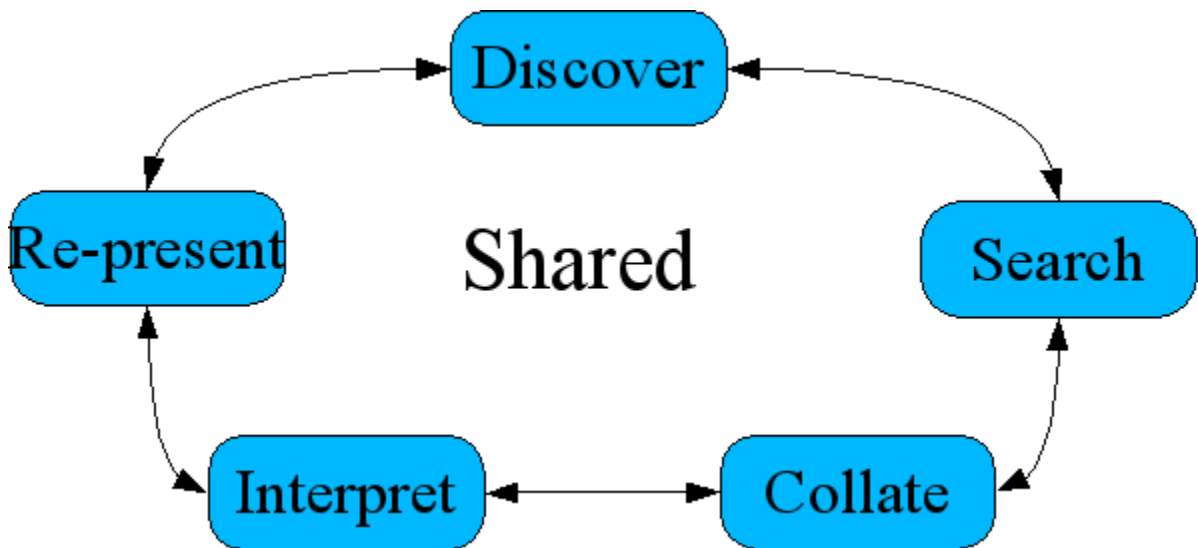


Abbildung 4.1: Workflow-Zyklus nach Paepke

Der Desktop-Client, das Programm mit dem Endnutzer in der Regel arbeiten, wurde nach den Gestaltungsgrundsätzen des WOB-Modells umgesetzt. Das WOB-Modell wird im folgenden Abschnitt erläutert.

4.3 WOB Modell

Das WOB-Modell ist ein softwareergonomisches Modell mittlerer Abstraktionsstufe zwischen den rechtsverbindlichen Normen für die Softwaregestaltung, wie die ISO 9000, und den Vorgaben der Style-Guides, die die Gestaltungsmittel auf der konkreten Implementationsebene regulieren. Die Vorgaben der Normen sind zu abstrakt um zu einer direkten Operationalisierbarkeit zu führen, darüberhinaus sind sie vielen Softwareentwicklern nicht ausreichend bekannt (Krause 1997b).

Die Regeln der Styleguides hingegen sind in sich widersprüchlich, da sie sich auf eine sehr konkrete Gestaltungsebene begeben, auf der Zielkonflikte in der Regel nicht erkennbar oder auflösbar sind.

Das WOB-Modell basierte ursprünglich auf einer Reihe von empirischen Untersuchungen an der Universität Regensburg. Es wurde im Rahmen von Projekten zur Entwicklung von Informationssystemen dort und am Informationszentrum Sozialwissenschaften in Bonn entwickelt und ausgebaut.

Die *Werkzeugmetapher* ist für das DAFFODIL-System mit seinen zahlreichen Services gut geeignet, weil spezielle Dienste auf dynamische Weise als Werkzeuge in den Arbeitsplatz des Benutzers eingebunden werden können. Durch die Konzepte des Modells kann trotz der Funktionsvielfalt eine konsistente und ergonomische Oberfläche angeboten werden.

Das Modell beinhaltet einen Katalog von Gestaltungsrichtlinien, welche im Folgenden beschrieben sind.

4.3.1 Generelle softwareergonomische Prinzipien

Um die auffälligsten Schwierigkeiten, die die Anwender von Informationssystemen in den empirischen Untersuchungen zu meistern hatten, abzumildern, basiert das WOB-Modell auf den folgenden generellen softwareergonomischen Prinzipien:

- *Dynamische Anpassung und kontextsensitive Durchlässigkeit*

Wird die Information einer Eingabe an anderer Stelle im System nochmals benötigt oder ist die Eingabe eines später in der Dialogleitlinie vorgesehenen Feldes deduktiv aus einer vorangegangenen Eingabe ableitbar, so wird das System selbsttätig diese Eingaben vornehmen. So wird einerseits der Benutzer von der Eingabe redundanter Informationen entlastet und er erhält intelligentes Feedback, ob die bisherigen Eingaben seiner Intention entsprechen.

- *Dialogleitlinien*

Leitlinien zur optimalen Abfolge von Dialogen und Formularen sollen helfen, den Nutzer zu führen. Dies muss jedoch abgewogen werden gegen die dadurch eingeschränkte Handlungsfreiheit.

Bei einer Aufteilung einer Aufgabenanalyse in einzelne Handlungsschritte werden jedem Schritt einzelne Fenster, bzw. Dialoge und Formulare, zugeordnet

und in eine möglichst „natürliche“ Reihenfolge gebracht. Da es jedoch in aller Regel eine solche natürliche Abfolge nicht geben kann und somit keine nutzerübergreifenden Abfolgemuster analytisch festzulegen sind, ist für den einzelnen Benutzer eine möglichst große Handlungsfreiheit bedeutsam.

Die Leitlinien müssen also auf einer Abstraktionsebene festgelegt werden, in der individuelle Unterschiede bei der Abarbeitung der Handlungsschritte möglich bleiben und dennoch klar ist, welche Schritte als Fortsetzungsmöglichkeit im aktuellen Status sinnvoll sind. Dies wird durch die Werkzeugmetapher unterstützt (siehe unten).

Die durch die Objektorientierung gegebene Flexibilität scheint zunächst die Forderung nach individueller Freiheit zu erfüllen und so eine unkomplizierte Lösung zu bieten. Es wird jedoch immer Unterschiede zwischen den Benutzern geben, so dass einige mehr Freiheit in der Benutzung und andere mehr Führung und Empfehlung brauchen. Das Modell empfiehlt somit die Analyse und Unterstützung sinnvoller Dialogabläufe, ohne die Freiheit der Abfolge mehr als nötig einzuschränken.

4.3.2 Iteratives Retrieval

Das WOB-Modell wurde als ergonomisches Leitmodell für Benutzungsoberflächen von Informationssystemen erstellt. Es geht von einem iterativen Retrievalmodell aus, bei dem nach der initialen Suche und dem Vorliegen der ersten Ergebnisse ein Zyklus mit Verfeinerung und Transformation der Anfrage die Regel und nicht die Ausnahme ist. Das iterative Retrieval wird durch drei Aspekte des Modells unterstützt:

- *Zustandsanzeige mit Korrekturmodus*

Alle bisher durch den Nutzer gemachten Einstellungen und Anfrageparameter, die Einfluss auf die Ergebnismenge haben, werden zusammen mit den Ergebnissen dargestellt. Die Information wird auf eine möglichst kompakte Weise dargestellt und bleibt editierbar, sodass schnell und unkompliziert die Anfrage modifiziert werden kann, um zu einer verbesserten Ergebnismenge zu kommen.

- *Query-By-Example-Modus*

Bei dem hier vorgeschlagenen Query-By-Example-Modus (QBE) können Textstücke als Anfrage verwendet werden, um ähnliche Dokumente zu finden. Auch ganze Texte oder Metadatenätze können als Suchanfrage verwendet werden.

- *Graphisches Retrieval*

Bei Faktendomänen, wie beispielsweise WING, dem Werkstoffinformationssystem, in dessen Kontext das WOB-Modell entwickelt wurde, bietet es sich an, Wertebereiche in Graphen abzubilden, die via direkter Manipulation Selektionen auf den gesuchten Wertebereich erlauben. So kann der Anwender seine Anfrage verfeinern, ohne den grafischen Modus zu verlassen. Auch in Bezug auf das Dokumentretrieval kann es sinnvoll sein, bestimmte Faktenrelationen oder Ähnlichkeitsbeziehungen in Form eines Graphen darzustellen, mit dessen Hilfe der Anwender den Informationsraum erschließen kann.

4.3.2.1 Modalitätsmischung

Von einer Kombination mit sanftem Überstieg zwischen verschiedenen Modalitäten können Anwender profitieren. Die empirischen Untersuchungen zum WING-Informationssystem haben gezeigt, dass ein Wechsel vom Formularmodus in den grafischen Anfragemodus den Anwendern natürlich erschien. Die Möglichkeit, in einem Graphen ihre Anfrage zu verfeinern, kam den Bedürfnissen der Benutzer entgegen. Das Bezugnehmen auf die grafische Darstellung und die Reformulierung direkt im Graphen wurde besser angenommen, als die Rückkehr in den Formularmodus oder die natürlichsprachliche Anfrage:

„Liegt ein erstes Ergebnis vor, das sich der Benutzer als Liniengraphik anzeigen lässt, „denkt“ der Benutzer mit dieser Kurve weiter: „Das ist schon ganz schön, aber die Kurve des Elastizitätsmoduls müsste im niederen Temperaturbereich etwas höher liegen“. Das bedeutet, dass die Liniengraphik selbst als Modus zur Formulierung der weiteren Suchabsicht benutzt wird. Der Zwang, die Suchabsicht wieder als Parametermenge in der strukturierten Umgebung des grafisch-direktmanipulativen Ersteinstiegs zu formulieren, ist in dieser Situation genauso „unnatürlich“ wie der natürlichsprachliche Abfragemodus. Der Modus der Ergebnisdarstellung bestimmt die für die Weiterführung der Suche optimale Modalität.“ (Krause 1997b, S. 13)

Somit wirkt eine Modalitätsmischung positiv, die Wahlfreiheit ist jedoch durch die kognitiven Gegebenheiten eingeschränkt. Die Modalitäten sind also nicht uneingeschränkt mischbar.

4.3.2.2 Zustandsanzeige mit Korrekturmodus

Die Zustandsanzeige kann je nach Kontext verschiedene Ausgabemodi beherrschen:

- *Natürlichsprachliche Generierung*

Hierbei wird die Paraphrase der Suchanfrage als natürlichsprachlicher Satz ausgegeben. Die Benutzer werden so an die natürlichsprachliche Dialogmöglichkeit herangeführt und können durch Ändern der Ausgabe Varianten der Anfrage absetzen. Dieses Konzept wurde im Werkzeuginformationssystem WING eingesetzt. Siehe auch (Marx 1996).

- *Formalsprachliche Generierung*

Bei der formalsprachlichen Generierung wird ein Anfragestring in einer oder mehreren Anfragesprachen ausgegeben und kann vom Anwender modifiziert werden. Kennt der Anwender die Anfragesprache, hat dies den Vorteil der logischen Präzision und der eindeutigen Semantik. Die Lernkurve ist jedoch für mit der formalen Sprache nicht vertraute Nutzer höher als im natürlichsprachlichen Fall.

- *Komprimierter Eingangsbildschirm*

Beim komprimierten Eingangsbildschirm werden die Formularbestandteile des ausführlichen Eingangsbildschirms in kompakter Weise genutzt, um den Zustand der Anfrage mit allen notwendigen Bestandteilen wiederzugeben. Syntaktische Ausdrucksmöglichkeiten werden bei dieser Gestaltung durch visuelle Formalismen ersetzt.

Nachdem das Eingangsformular strikt selbsterklärend ausgelegt war, ist die komprimierte Darstellung der Suchparameter beim iterativen Verfeinern der Anfrage Merkhilfe, Zustandsanzeige und Eingabefläche.

4.3.3 Strikte Objektorientierung

Das WOB-Modell verzichtet auf Menüstrukturen, da diese ab einer gewissen Komplexitätsstufe nicht mehr skalieren und zu erheblichem Such- und Interpretationsaufwand führen. Als Ersatz ist eine konsistente und strikte Objektorientierung mit reinem Objekt-Objekt-Interaktionsschema vorgesehen – das bedeutet, dass Funktionsobjekte durch direkte Interaktion auf Datenobjekte wirken.

Diese Gestaltungsweise hat Vorteile in Bezug auf die Erweiterbarkeit: Jedes Objekt definiert seine Funktionen und Parameter. Es gibt keine Menüstruktur, die durch das Hinzufügen eines Objektes inkonsistent werden könnte. Es ist jedoch Vorsicht geboten:

„Funktionsobjekte lassen sich nur dann problemlos einsetzen, wenn das Bild des (Ein)legens/Einschiebens des Objekts auf/in das Funktionsobjekt in der realen Welt eine Parallele hat und hohe Eindeutigkeit der Funktionszuordnung gegeben ist (in diesem Sinn positiv: Drucken, Dokument in Schrank ablegen, Dokument auf Kopierer ziehen usw.). Zudem muss die Richtung der Aktion stimmen. Auch in der realen Welt soll das Objekt auf das Funktionsobjekt zu bewegt werden und nicht umgekehrt. In diesen engen Grenzen erwiesen sich Funktionsobjekte bei den empirischen Tests als ein wesentlicher Bestandteil direktmanipulativer Benutzungsoberflächen, die die Selbsterklärungsfähigkeit erhöhen.“ (Krause 1997b, S.40)

Die Abstraktionsebene eines Werkzeuges in Bezug auf seine Funktionsvielfalt muss sorgsam gewählt werden. Wenn das Werkzeug nur eine Funktion hat, kann es leicht passieren, dass der Anwender viele Instanzen mit unterschiedlicher Parametrisierung benötigen würde, um alle Anwendungsfälle abzudecken. Krause nennt als Beispiel die Schreibstift-Objekten die in der strikt objektorientierten Bürosoftware N/-JOY eingesetzt wurden, um verschiedene Schriftstile zu repräsentieren. Hat das Werkzeug hingegen zu viele Funktionen, wird es unübersichtlich und die Wirkweise kann nicht mehr intuitiv durch die Metaphorik verstanden werden.

4.3.4 Werkzeugmetapher und doppelte Interpretierbarkeit

Das *Werkzeug* ist die zentrale Metapher des WOB-Modells. Funktions-Objekte werden durch Ikonen symbolisiert. Dazu gehören im geöffneten Zustand Fenster, bzw. Dialoge. Diese Werkzeuge lassen sich, wie die realen Vorbilder, einstellen, also über einen Einstellungsdialog parametrisieren. Über Ziehen und Fallenlassen (*Drag and Drop*) lassen sich andere Objekte der Oberfläche auf das Werkzeug bewegen und dieses wird dann mit den aktuellen Einstellungen auf dieses Objekt angewandt.

Im Vergleich zum herkömmlichen Objekt-Funktions-Schema, das etwa einer Menüführung zugrunde liegt, dreht sich dabei die Richtung der Handlung um. Statt ein Objekt zu selektieren und aus dem Menü eine Funktion auszuwählen, die darauf

anzuwenden ist, selektiert der Anwender ein Datenobjekt und führt dieses einem Funktionsobjekt zu. So manipuliert er die Datenobjekte direkt, um Funktionen anzuwenden.

4.4 Architektur, Agenten und Dienste

DAFFODIL ist als verteiltes System konzipiert – das bedeutet, die Softwarekomponenten, die zusammen das DAFFODIL-System bilden, können sich auf unterschiedlichen Computersystemen befinden und tauschen Nachrichten und Ereignisse über Netzwerkverbindungen aus. Eine offene Menge von Diensten und verteilten Komponenten teilen sich dabei eine nachrichten- und ereignisbasierte Infrastruktur. Die meisten der Komponenten gelten dabei als „Agent“, da sie autonom und kontinuierlich laufen und auf Anfragen ein definiertes Ziel (Goal) zu erreichen versuchen – in der Regel senden sie dann zu einem späteren Zeitpunkt eine entsprechende Nachricht. Sie stellen wiederum Anfragen an weitere Agenten und Dienste, die sie mit Unteraufgaben betreuen, sodass sich die Bearbeitung einer Aufgabe in einer Kaskade von asynchronen Nachrichten zur Delegation von Teilaufgaben über das Agentennetz ausbreitet¹.

Grob lassen sich drei konzeptuelle Klassen von Komponenten einteilen:

- *Wrapper*

Die Wrapper-Dienste ermöglichen den Zugriff auf diverse heterogene Datenquellen. Darunter sind lokal verfügbare Datenbanken sowie entfernt verfügbare Web-Services oder Internetseiten, die eine Anfrage per Formular erlauben. Die Wrapper haben eine gemeinsame Anfragesprache, sodass die Klienten einheitlich Anfragen an Gruppen von Wrappern verteilen können. Wrapper müssen regelmäßig gepflegt sein, da sich die angesprochene Datenquelle verändern kann – dies gilt insbesondere für Web-basierte Dienste. Zur Vereinfachung der Erstellung und Pflege von DAFFODIL-Wrappern existiert ein spezielles Toolkit.

- *Dienste*

Die Middleware-Dienste bieten Funktionen und Daten an, die zur Realisierung von Stratagemen und Taktiken notwendig sind. So gibt es zum Beispiel Dienste zum Anreichern der Metadaten eines Dokuments quer über verschiedene

¹Zu der Zeit als DAFFODIL konzipiert wurde war der Begriff „Service Oriented Architecture“ (SOA) noch nicht geprägt – inzwischen würde man eine solche Struktur so bezeichnen.

Wrapper hinweg, und es gibt auf Autoren, Journale und Konferenzen spezialisierte Dienste

- *User-Interface-Clients*

Das Programm, mit dem die meisten Anwender die Daffodil Dienste nutzen, ist die oben beschriebene Benutzungsoberfläche, in der eine Reihe von Werkzeugen integriert ist, um die diversen Dienste anzusprechen. Ebenso können Dienste in der Oberfläche verankert sein, die auch von der Backend-Seite aus angesprochen werden können.

Die grundlegende technische Architektur ist in Abbildung 4.2 dargestellt. Die konzeptuelle Schichtung der Dienste nach Move, Taktik, Stratagem und Strategie, welche sich an den Bates-Konzepten orientiert, ist in Abbildung 4.3 zu sehen.

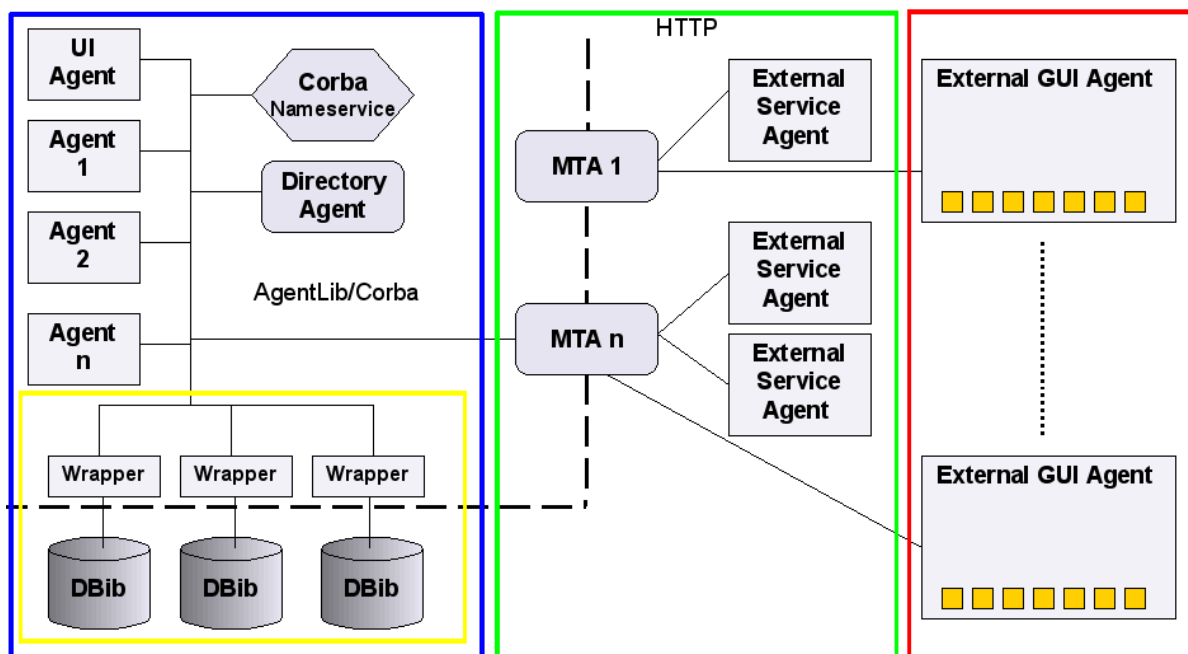


Abbildung 4.2: Verteilte Architektur von Daffodil

Details zu der verwendeten Agenteninfrastruktur finden sich bei Müller (2001). Eine weitergehende Erläuterung der Architekturentscheidungen in Hinblick auf die strategische Unterstützung beschreibt Klas (2007).

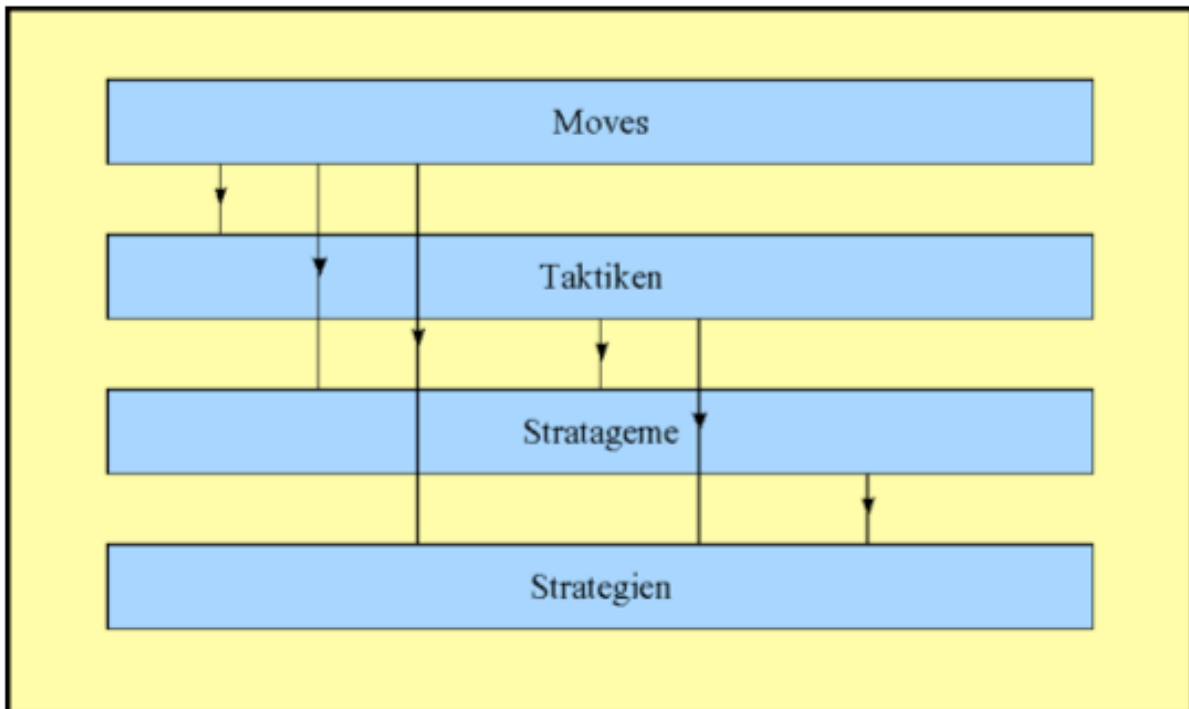


Abbildung 4.3: Konzeptuelle Schichtung der DAFFODIL-Dienste

4.4.1 Modularität

Durch die verteilte Dienstarchitektur und den Grundgedanken eines agenten-gestützten Informationssystems ist DAFFODIL ganz grundsätzlich keine geschlossene Anwendung mit fest umrissenen Eigenschaften und Funktionsumfang. Deshalb gehörte zur Grundkonzeption die Anforderung der Modularität und Erweiterbarkeit.

Die Metapher der Arbeitsumgebung (Desktop), die die Werkzeuge zum Finden, Aufbewahren und Wiederverwenden von Informationsobjekten zusammenführt und gemeinsam verwendbar macht, ist der Schlüssel, diese Anforderung zu erfüllen.

Ein Werkzeug stellt in Daffodil ein klar abgegrenztes Objekt dar, das Interaktion des Anwenders, die Eingabe von Informationsobjekten und das Senden und Empfangen von Nachrichten unterstützt. Kein Werkzeug ist direkt von der Existenz oder der Funktionsweise anderer Werkzeuge abhängig. Jedes Werkzeug könnte auch alleine in einem ansonsten leeren DAFFODIL-Desktop verwendet werden.

DAFFODIL wurde im Rahmen von Diplomarbeiten um neue, spezialisierte Werkzeuge erweitert, etwa den Multi-Level-Hypertext-Browser von (Chojnacki u. Fischer 2005).

4.4.2 Nachrichtenarchitektur

In diesem Abschnitt wird die horizontale Kommunikation zwischen den Agenten und deren Protokolle bzw. deren verwendete Nachrichten erläutert. Protokolle definieren eine standardisierte Abfolge von Nachrichten und machen damit eine Konversation zwischen Agenten über mehrere Nachrichten hinweg möglich. Hierzu werden *Performative* zur Definition von Protokollen benötigt.

Eine Nachricht wird mittels der Methode `send` an andere Agenten verschickt. Der Aufbau einer Nachricht besteht aus folgenden Parametern:

Performative geben den Nachrichtentyp an (siehe nächster Abschnitt).

Sender gibt den Absender der Nachricht an.

Empfänger gibt den oder die Empfänger der Nachricht an.

Request-ID ist eine eindeutige ID, die vom Sender erzeugt oder übernommen wird.

Inhalt enthält die eigentliche Nachricht im XML-Format.

Performative

Die Performative dienen dazu, bestimmte Protokolle von Nachrichten einzuhalten und als Unterscheidungskriterium.

Die folgenden Performative sind bisher im System festgelegt, ein Hinzufügen von neuen ist jederzeit möglich:

INIT Das Performativ `INIT` wird genutzt um den Anfang einer Konversation zu markieren.

ASK Nachrichten mit diesem Performative stellen Anfragen an den Empfänger.

TELL beinhaltet Nachrichten, die entweder Antworten auf Anfragen des Typs `ASK` sind oder einfach nur Informationen enthalten.

ACK dient der Bestätigung einer vorherigen Nachricht.

UNKNOWN besagt, dass die vorherige Nachricht nicht verstanden worden ist.

CFP (Call for Proposal) leitet das Kontrakt-Netz-Protokoll ein

Die ereignisbasierte Nachrichtenarchitektur zwischen den Werkzeugen der Oberfläche macht durch eine Vermittlungsstelle (*Message Transfer Agent* (MTA)) ebenfalls von

dieser systemübergreifenden Nachrichtenstruktur Gebrauch. Interne Ereignisse, die einem ASK- oder TELL- Ereignis entsprechen, werden in konforme Nachrichten gewandelt und per HTTP an die Dienste des Agentennetzwerkes gesendet. Die Antworten werden anhand der Request-ID dem ursprünglichen Sender innerhalb der Oberfläche zugestellt. Der Nachrichtenaustausch im GUI-Client wird im Anhang B erläutert.

4.4.3 Interaktionsschemata

Um den Anwendern bei der Recherche eine möglichst adäquate Schnittstelle anzubieten, wurden die Interaktionsschemata anhand der folgenden zentralen Szenarien ausgestaltet:

1. Ausgehend von den Autoren wird nach weiteren Dokumenten derselben Autoren, ihrer Ko-Autoren oder nach Dokumenten zentraler Autoren in der Nähe der Autoren gesucht.
2. Ausgehend vom Dokumenttitel wird nach weiteren Dokumenten mit denselben Begriffen oder nach Dokumenten mit semantisch breiteren oder engeren Begriffen (mit Hilfe des Thesaurus) im Titel, Schlagwort oder Abstract gesucht.
3. Ausgehend vom Zeitschriften- oder Konferenztitel werden weitere ähnliche Dokumente in der selben Zeitschrift oder Konferenz gesucht, evtl. über mehrere Jahrgänge. Diese Vorgehensweise entspricht dem *Journal-Run*-Strategem (siehe Abschnitt 4.5.4).

Hier wird jeweils ausgehend von bekannten Informationsobjekten iterativ vorgegangen, um weitere Informationsobjekte zu erreichen. Die in einem Werkzeug vorhandenen Informationsobjekte werden in einem weiteren Verarbeitungsschritt als Eingabe genutzt, um ein Ergebnis (Funktion) zu erhalten. Dies wird in Interaktionen mit Funktionsobjekten abgebildet.

- *Direkte Manipulation*

Zum Aktivieren dieser Ähnlichkeitssuchen ist es möglich, die Informationsobjekte via *Drag and Drop* auf das entsprechende Werkzeug zu ziehen.

Die direkte Manipulation von visuellen Darstellungen, etwa einem Netzwerk von Koautoren, soll das explorative Vorgehen bei der iterativen Recherche erlauben und kognitiv vereinfachen.

- *Kontextmenüs*

Daneben wird beim Ausgangspunkt der jeweiligen Interaktion ein Kontextmenü mit diesen Funktionen angeboten. Diese Aktion versendet eine Nachricht an ein oder mehrere Funktionsobjekte, die mit dieser Eingabe eine Recherche oder Transformation durchführen können.

- *Verknüpfungen*

Eine mit der Detailansicht eines Dokumentes (Metadatenansicht) verknüpfte Funktion wird angeboten, um von einem relevanten Treffer zu weiteren relevanten Treffern zu navigieren. Hier werden Links und Buttons eingebettet, die die entsprechende Funktion anstoßen.

4.5 Aufgaben und Szenarien für Agenten im Daffodil Kontext

Nach dem kurzen Überblick über die DAFFODIL-Konzepte werden im Folgenden die Szenarien und Anwendungsfälle für Agenten in DAFFODIL diskutiert. Es wird jeweils Bezug genommen auf die vorgestellten Probleme bezüglich der ergonomisch sinnvollen Aufgabenteilung.

4.5.1 Föderierte Suche

Die Zusammenführung verschiedener Datenquellen zu einer integrierten föderierten Suche ist die Grundlage für die Unterstützung des Recherchevorganges. Mit der offenen und verteilten Architektur ermöglicht DAFFODIL eine solche föderierte Suche.

DAFFODIL bietet bereits ein breites Spektrum an Werkzeugen und Diensten, die dem Anwender bei der Anfrageformulierung helfen können: Einen Thesaurusbrowser, einen Klassifikationsbrowser und einen Dienst zum Finden von themenbezogenen Begriffen. Die proaktive Unterstützung der Eingabe, verbunden mit Fehlerkorrektur und kontextbezogenen Vorschlägen zur Anfrageergänzung kann diese bereits vorhandenen Werkzeuge sinnvoll einbinden.

Die Leistungsmerkmale des DAFFODIL-Agentensystems auf dieser Ebene sind:

- *Einheitliche Suchmöglichkeit*

Es wird eine einheitliche Suchmöglichkeit erreicht, unabhängig von der Quellenauswahl. Es steht ein quellenübergreifendes Metadatenschema zur Verfügung, unabhängig von den Schemata der darunterliegenden Quellen. Ein Dienst kann dabei auf sekundäre Dienste zurückgreifen, etwa zur Query-Transformation, wenn es nötig ist.

- *Zusammenführung der Resultate*

Die Zusammenführung der verschiedenen Ergebnisse aus unterschiedlichen Quellen zu einer Ergebnismenge. Dabei wird die Resultatliste von Duplikaten bereinigt und mit Annotationen versehen, ob ein Dokument schon einmal angesehen oder in die persönliche Handbibliothek des Anwenders gespeichert wurde.

- *Integration der Metadaten*

Die Anreicherung von Metadaten zu einem gegebenen Datensatz aus mehreren Quellen. Wenn ein Artikel in verschiedenen Quellen gefunden wurde, werden die Felder der Metadaten, basierend auf dem homogenisierten Datenschema, zusammengeführt.

- *Relevanzberechnung*

Unterstützung einer Relevanzberechnung. Da nicht alle Quellen Relevanzwerte liefern und eine sinnvoller Normalisierung vorhandener Werte nicht möglich ist, ohne die dahinterstehenden Algorithmen zu kennen, müssen die Ergebnisse nach der Zusammenführung zu einer Ergebnisliste neu bewertet werden.

Hierzu werden Methoden des Information Retrieval auf die verfügbaren Metadaten angewendet und Relationen, wie beispielsweise Koautorenbeziehungen in Betracht gezogen.

Die agentengestützte Zusammenführung heterogener Informationsquellen bietet an sich schon einen Vorteil bei der Recherche, dennoch gehen durch die indirekte Anbindung an die Datenquellen auch Möglichkeiten und Informationen, wie etwa ein einheitliches Ranking verloren. Zum Teil lassen sich diese Nachteile durch zusätzliche stützende Dienste ausgleichen. Eine föderierte Suche² dauert jedoch immer mindestens so lange, wie die langsamste Informationsquelle antworten kann, zuzüglich einem Aufwand, der durch die genannten Mechanismen zur Zusammenführung und Nachbearbeitung der Ergebnisliste anfällt – dennoch zeigt die Evaluation in Ab-

²vergleiche auch Abschnitt 3.5.2

schnitt 5.2, dass der entstehende Mehrwert bei komplexeren Rechercheszenarien Vorteile für die Gesamtrechenzeit hat.

Demnach bedeutet eine föderierte Suche ein großes Potential für den Anwender, ist für sich alleine genommen aber noch keine eindeutige Verbesserung gegenüber den einzelnen, nebenständigen Informationsangeboten.

Die hier vertretene These ist, dass erst die tiefgehende Verknüpfung der Information auf den verschiedenen Handlungsebenen und der dadurch erzeugte echte Mehrwert, zu der gewünschten Verbesserung der Situation für den Suchenden führt.

4.5.2 Agentengestützte Personalisierung und Adaptivität

Neben der Heterogenität der Quellen spielen für die Anwender auch ihre speziellen Herangehensweisen an die Rechercheproblematik und die dadurch bedingten Vorlieben eine große Rolle für den Erfolg. Wie in Abschnitt 3.1.1 diskutiert, soll ein entscheidender Mehrwert von Software-Agenten durch Adaptivität – durch Beobachtung des Anwenders und Anpassung auf seine speziellen Vorlieben und Bedürfnisse – entstehen. In (Shneiderman u. Maes 1997) nennt Maes als Beispiel die Film-Recherchehilfe Firefly (siehe Abbildung 4.4), die sie dem direktmanipulativen und auf Visualisierung basierendem FilmFinder (siehe Abbildung 4.5) von Shneiderman gegenüberstellt.

Ein aktuelles Beispiel dieser Art von *adaptivem Recommendation-Dienst*, analog zu Firefly, erfreut sich derzeit großer Beliebtheit: Das Musik-Recommendation-Portal *Last.fm*³. Dieser Dienst beruht darauf, dass sich Anwender, die dort ein Profil anlegen, einen Agenten in ihre Musikbibliothek einbinden. Dieser meldet die Abspielreihenfolge des Anwenders – samt Metadaten – an den Last.fm-Dienst (das nennt last.fm *scrobblen*), der aus den Daten ein statistisches Profil errechnet. Dieses Profil wird mit dem Profil anderer Anwender abgeglichen. Bei hoher Übereinstimmung gibt es Match-Making-Listen, mit den nächsten Nachbarn mit ähnlichem Musikgeschmack. Aus deren Profilen wiederum werden spezielle Empfehlungen zum Hineinhören errechnet und als „Radio“-Sender angeboten. Auf diese Weise kann ein musikinteressierter Hörer leicht neue Interpreten, Alben und Gleichgesinnte „Nachbarn“ finden.

Von Ostwinkel (2002) werden im Rahmen einer Diplomarbeit für die Verwendung von *Adaptivität* in DAFFODIL folgende Ansatzpunkte genannt:

³<http://last.fm>

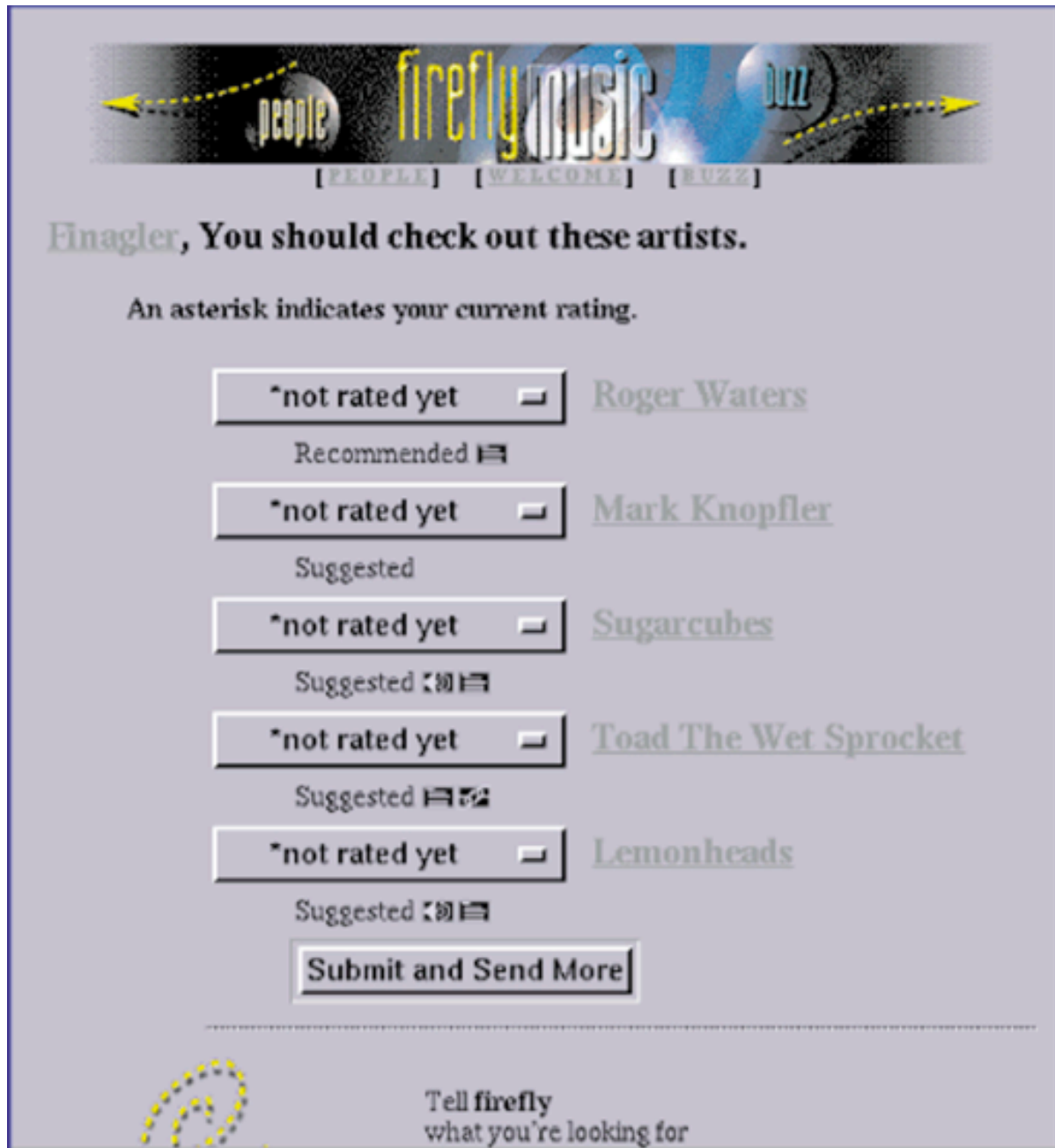


Abbildung 4.4: Firefly von Maes – adaptive Recommendation.

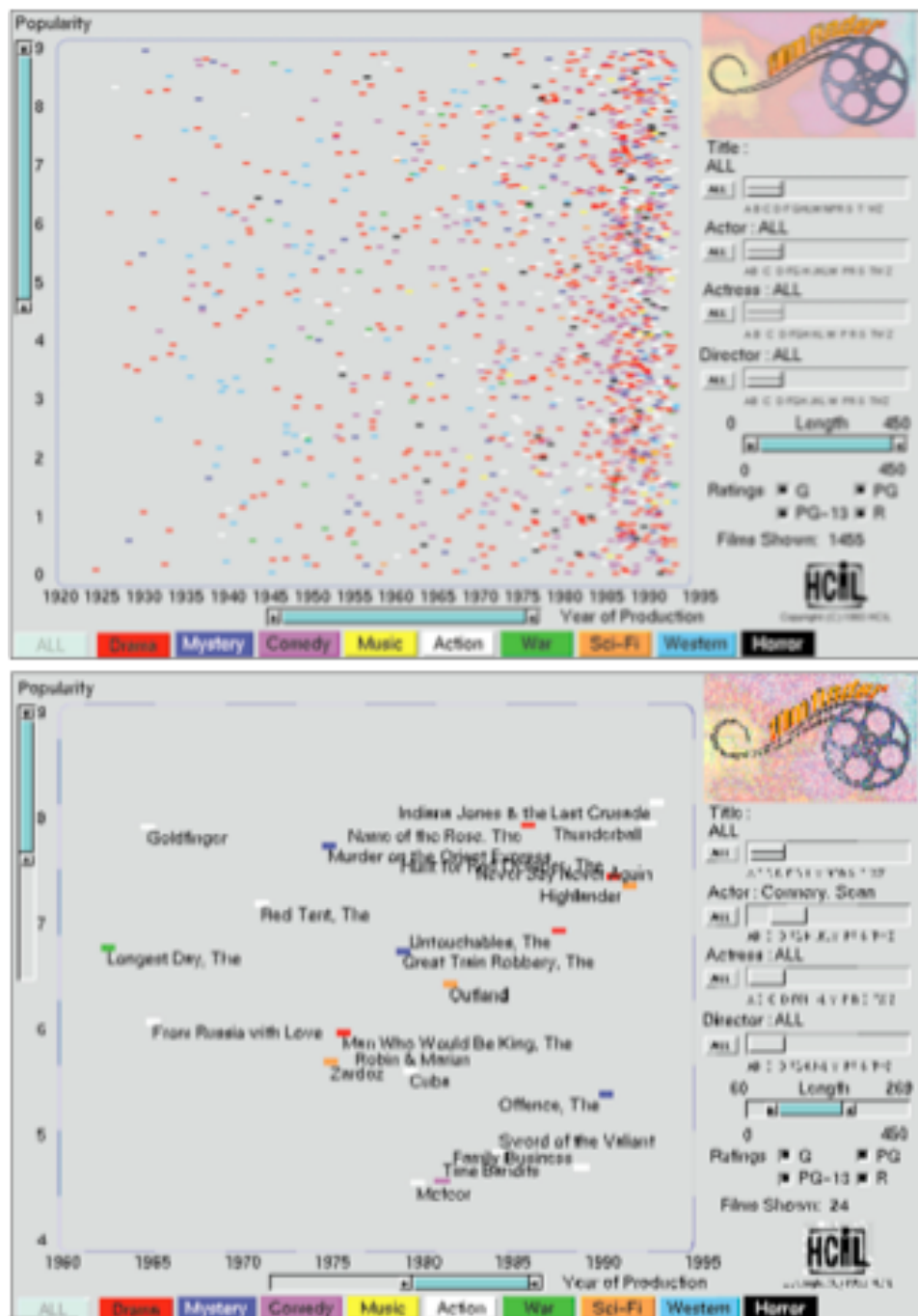


Abbildung 4.5: Shneidermans FilmFinder – direktmanipulative Alternative zu Firefly, um interessante Medien zu finden.

- *Adaption von Systemparametern*

Die internen Abläufe in DAFFODIL sind komplex und hängen von einigen Parametern ab, die sich nicht gut im Vorhinein bestimmen lassen. Ein Beispiel ist die Fehlererkennung und Fehlertoleranz, insbesondere an den Systemübergängen zu externen Informationssystemen, den Wrappern. Hier kann sich vieles ändern, wodurch eine Reihe von Funktionen betroffen sein können, die sich auf die Dienste des Wrappers verlassen. Wrapper können Fehlerzustände melden oder beobachtbar machen und so eine Wahrnehmung der geänderten Außenwelt und damit auch eine Anpassung auf diese Änderungen ermöglichen. Die Anpassungen können von einer Anpassung von Time-Outs bis zu einer Abschaltung von nicht mehr funktionierenden Diensten und entsprechenden Meldungen an das Monitoring und die Systemadministratoren reichen.

Ein anderes Szenario ist die für den Anwender im Rahmen der föderierten Suche bedeutsame durchschnittliche Antwortzeit und allgemeine Antwortwahrscheinlichkeit je Datenquelle. Die Antwortzeiten können je nach Tageszeit und Auslastung schwanken. Ebenso hängt die Antwortzeit von der logischen Struktur einer Anfrage ab. Das Lernprofil soll für einzelne Zeitfenster (z.b. 12:00 Uhr – 13:00 Uhr) die mittlere Zugriffszeit und die durchschnittliche Fehlerquote umfassen. Zudem schlägt Ostwinkel vor, die Anfragetypen zu unterscheiden (es gibt Metadatenanfragen nach Autor, Titel, Jahr oder Volltextanfragen, sowie spezielle Anfragen nach Journal-/Konferenz oder Zitationsrelationen in Daffodil), um ein möglichst differenziertes Profil zu erhalten.

- *Adaption auf Eingabemuster*

Da DAFFODIL als offenes System von Werkzeugen für viele Anwender mehr an Funktionalität anbietet, als vom einzelnen auf Dauer genutzt werden kann, schlägt Ostwinkel vor, die Eingabemuster je Anwender in ein spezifisches Nutzungsprofil umzurechnen und die Oberfläche an die spezifischen Bedürfnisse anzupassen, indem nicht verwendete Werkzeuge versteckt werden oder häufig verwendete in eine sinnvolle räumliche Nähe zueinander gerückt werden.

- *Erlernen optimaler Suchpfade*

Nach dem Pfad-Ansatz von Chalmers (1999) soll eine Abfolge von Informationsobjekten, welche ein Anwender betrachtet hat, als Pfad verstanden werden. Werden gleiche, oder ähnliche Informationsobjekte wiederholt in ähnlichen Kontexten nacheinander, bzw. im selben Pfad, betrachtet, eignen sich diese Objekte möglicherweise als Vorschlag für andere Anwender in ähnlichen Situationen.

- *Erlernen von Indikatoren für Stratagem-Schwellenwerte*

Ein Stratagem war eine spezielle Teilrecherche um für ein Thema möglichst umfassende Information zusammenzutragen, etwa indem alle Ausgaben eines bestimmten Journals oder einer Konferenz zu diesem Thema durchgesehen werden, oder alle Veröffentlichungen eines Autors, bzw. seines kollaborativen Umfeldes recherchiert werden. Ausgehend von den Metadaten einer Ergebnisliste soll erkannt werden, wie vielversprechend die Anwendung einzelner Strategeme wäre, beispielsweise aufgrund der Häufigkeit eines bestimmten Journal-Titels.

Ostwinkel hat für diese Szenarien die Parameterstruktur, die Nachrichtenprotokolle und auch die Regeln zum Er- und Verlernen der Einträge in die Wissensbasis spezifiziert. Aufgrund der beschränkten Ressourcen dieser Diplomarbeit kam es jedoch nicht zu einer vollständigen Umsetzung in DAFFODIL.

4.5.3 Eingabeunterstützung und Anfrageunterstützung

Eine sinnvolle und gültige Anfrage zu formulieren, die möglichst akkurat das Suchbedürfnis widerspiegelt, ist gerade am Beginn eines Recherchevorgangs eine ernstzunehmende Hürde. Da sich die Anwender aber gerade zu diesem Zeitpunkt häufig in einem unsicheren Zustand befinden und noch nicht genau wissen, wie sie ihr Infor-

mationsbedürfnis exakt beschreiben können (siehe Abschnitt zu Belkins ASK-These in 3.2.3), brauchen sie hilfreiche Hinweise zur Unterstützung.

Um bereits zu diesem Zeitpunkt, also vor der ersten Suchanfrage oder auch während der Eingabe, sinnvoll Hilfestellung leisten zu können, braucht es effiziente Methoden auf der Service-Ebene. Es geht darum, Vorlageleistungen erbringen zu können, die einen engen Bezug zum Kontext des Benutzers haben. Die Berechnung der passenden Vorlagen muss aber zeitnah geschehen, um vom Anwender bemerkt und akzeptiert zu werden.

In einer Fallstudie (Schaefer u. a. 2005; Jordan 2005) wurde ermittelt, welche vorrangigen Probleme bei der Anfrageformulierung auftreten und ein Konzept vorgestellt, wie es möglich ist, auf proaktive und adaptive Weise solche Hilfestellungen und Vorlagen anzubieten. Durch die im vorhergehenden Abschnitt diskutierten Nachteile einer föderierten Metasuchmaschine ist es wichtig, schon die erste Suchanfrage möglichst gut und fehlerfrei zu formulieren. Dabei sind die Voraussetzungen für eine effektive Anfrage:

- Treffende, nicht zu hochfrequente Suchterme
- Korrekte Orthografie
- Korrekte Logik der Verknüpfung einzelner Anfragebedingungen
- Keine sich widersprechenden UND-verknüpften Anfrageteile
- Korrekte Annahmen zu Anfragesprache, Retrievalmechanismen und indexierten Dokumenten.

In diesen Fällen bietet sich eine aktive Anfrageunterstützung an, um die wichtigsten Fehler zu minimieren. Die entsprechende Umsetzung wurde zunächst mit Designstudien begleitet, um die Akzeptanz und Erwartungskonformität schon zu Beginn in den Entwicklungsprozess einfließen zu lassen.

Die Phase, in der die Suchenden beginnen, ihre Anfragen zu formulieren, ist nach den bisherigen Überlegungen besonders kritisch. Sie sollte entsprechend frei von unerwarteten Störungen sein und gleichzeitig ein Maximum an willkommener Unterstützung bieten.

In Abbildung 4.6 wird dem Benutzer ein zu seiner Eingabe passender Eintrag aus der Anfragehistorie präsentiert. Beobachtungen im Vorfeld haben gezeigt, dass Benutzer häufig Anfragen stellen, die in ähnlicher Form bereits in ihrer Historie auftauchen.

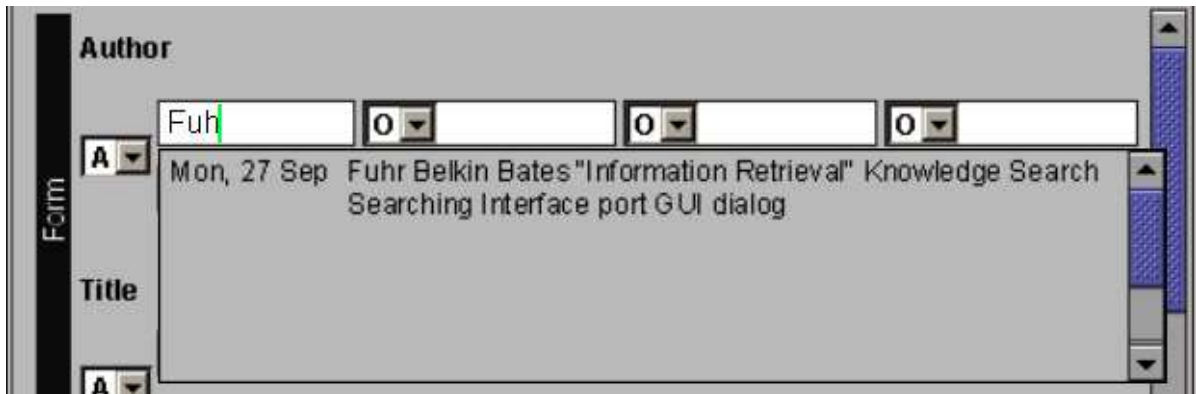


Abbildung 4.6: Vorschläge aus der Query-Historie

Beispielsweise weil sie sich noch vage erinnern, dass sie in einem bestimmten Zusammenhang bereits eine relevante Information gefunden hatten, aber versäumt haben, dieses Resultat dauerhaft in ihrer persönlichen Ablage zu sichern. Die Vorlage unterstützt in diesem Szenario die Erinnerung und hilft, auf die Anfrageformulierung aus der Historie schnell zurückzugreifen. Auch zur Modifikation schon gestellter aber nicht vollständig zufriedenstellender Anfragen eignet sich der Ansatz.

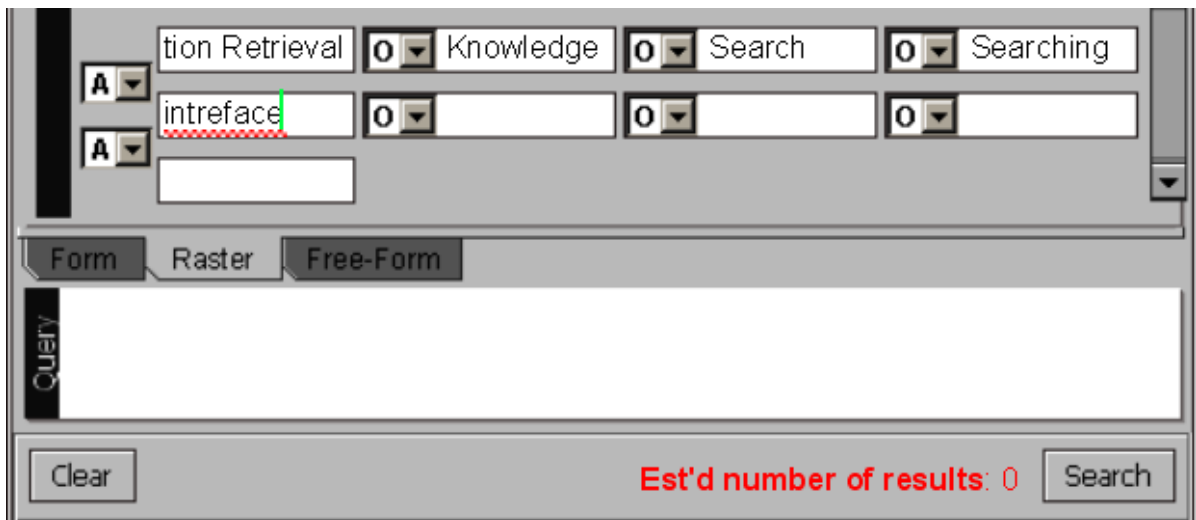


Abbildung 4.7: Rechtschreibprüfung

Eine Basisanforderung für eine effektive Anfrage ist, wie oben erläutert, die korrekte Schreibweise der Suchterme. Gerade durch die Unsicherheit der Nutzer am Anfang einer Recherche sind jedoch Tippfehler keine Seltenheit. Da Anfragen in einer Metasuchmaschine wie Daffodil aber lange dauern können, ist es für den Anwender frustrierend wegen eines Tippfehlers unnötige Wartezeiten in Kauf zu nehmen. Ent-

sprechend wird zum Eingabezeitpunkt die Eingabe auf eventuelle Tippfehler geprüft und entsprechende Markierungen angebracht. Die Art der Markierung ist in Abbildung 4.7 zu sehen.

Abbildung 4.8: Overconstrained Queries

In Abbildung 4.8 wird durch Markierungen in der Eingabemaske und durch Hinweise verdeutlicht, dass die derzeitige Anfrage logisch überspezifiziert ist. Das bedeutet, die Anfrage würde ohne Ergebnisse bleiben. Das Feststellen dieses besonderen Fehlerfalls stellt hohe Ansprüche an die Dienste von Daffodil, die schon zum Eingabezeitpunkt asynchron im Hintergrund prüfen müssen, ob die Anfragebedingungen noch erfüllbar sind. Zu diesem Zweck muss die Datenbank, welche die Autorenbeziehungen verzeichnet, stets aktuell sein und effizient antworten. Hier ist der Ansatzpunkt, die Pflege und das Aktuell-Halten der notwendigen Datenbasis an entsprechende Agenten zu delegieren. Die Agenten können sowohl die verschiedenen Wrapper-Dienste nutzen, um Kooperationsbeziehungen zwischen Autoren zu erheben, sie können aber ebenso die Datenströme analysieren, die jeweils bei Anfragen durch das Daffodil-System laufen und so als Seiteneffekt die Daten pflegen.

Da es sich hier aber zunächst um eine nicht-funktionale Design-Studie handelt, spielten die funktionalen Anforderungen an die Infrastruktur zunächst keine vorrangige Rolle. Es ging zunächst nur um Benutzerbefragungen und um die Akzeptanz der Gestaltung.

Nachdem aus einer Reihe von Gestaltungsvarianten diejenigen mit der besten Akzeptanz ermittelt waren, wurde ein Framework in Daffodil integriert, das flexible Prüfmechanismen und dynamische Vorlagen zum Eingabezeitpunkt ermöglicht.

Um einen guten Kompromiss zwischen Unaufdringlichkeit und Sichtbarkeit der Vorlagefunktionen zu erreichen, gibt Jordan (2005, S. 23f.) folgende nicht-funktionale Anforderungen als Ziel der Gestaltung an:

Die Mitteilungen der Funktionen an den Benutzer sollen:

ein gutes Timing haben: Der Benutzer sollte nicht zu lange warten müssen, bis ihm das System helfen kann, aber auch nicht dadurch abgelenkt werden, dass DAFFODIL zu häufig Vorschläge präsentiert.

zurückhaltend sein: Die proaktiven Funktionen dürfen nicht so auffällig sein, dass sie den Benutzer an der Arbeit hindern oder ablenken. Aufklappende Fenster, die sich über das gesamte Suchformular legen oder erfordern, dass der Benutzer Vorschläge des Systems per Copy-and-Paste in seine Anfrage übernimmt, scheiden damit z.B. aus.

bemerkt werden können: Die Funktionen sollten nicht so zurückhaltend sein, dass der Benutzer Mühe hat, sie zu bemerken.

willentlich provoziert werden können: Der Benutzer sollte die Möglichkeit haben, die Funktionen auch dann aufzurufen, wenn sie gerade nicht angezeigt werden – z.B. weil es unwahrscheinlich ist, dass die Vorschläge des Systems hilfreich sind.

übergangen werden können: Es sollte einfach für den Benutzer sein, Vorschläge zu übergehen, die der Benutzer nicht annehmen möchte.

Zum Beispiel: Aus dem Kontext, der sich aus der persönlichen Historie, sowie den in der Ablage gesammelten Dokumenten und Informationen zusammensetzt, wird eine Liste von Vorschlägen erstellt und angezeigt, wenn der Benutzer in einem Eingabefeld eine Pause einlegt.

Noch interessanter wird diese Form der *Recommendation*, wenn überindividuell, also

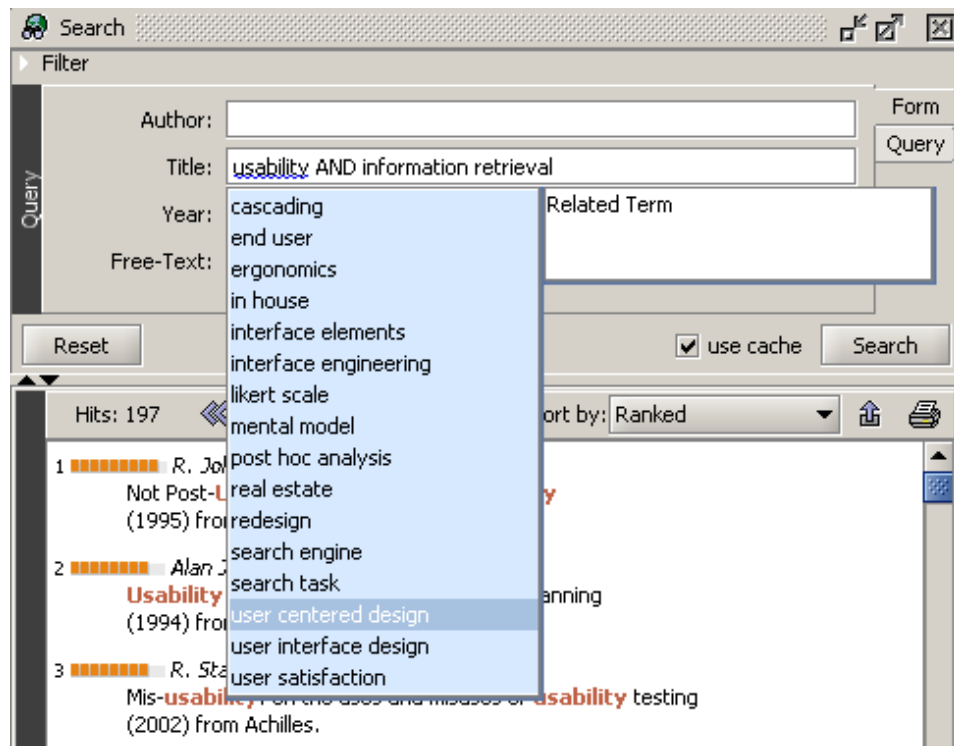


Abbildung 4.9: Vorlagen aus persönlichem oder überindividuellem Kontext

durch statistische Analyse vieler Benutzerprofile ermittelt wird, welche Anfragekomponenten mit hoher Wahrscheinlichkeit in Zusammenhang stehen.

4.5.4 Unterstützung der iterativen Anfrage und Resultatauswertung

Nach der ersten Anfrage bieten sich weitere Möglichkeiten, basierend auf der Ergebnismenge Vorlagen für weitere Anfragen und Navigationshilfen anzubieten:

- Aus den vorliegenden Resultaten können die vertretenen Autoren sortiert nach Häufigkeit oder nach der Anzahl der Kollaborationspartner aufgelistet werden. (Siehe Abbildung 4.10)
- Häufig vertretene Journale können als Ansatzpunkt für einen Journal-Run, also einer erschöpfenden Tiefensuche in der betreffenden Zeitschrift, dienen.
- Häufige Terme können aus den Titeln und Abstracts (falls verfügbar) extrahiert und angeboten werden.

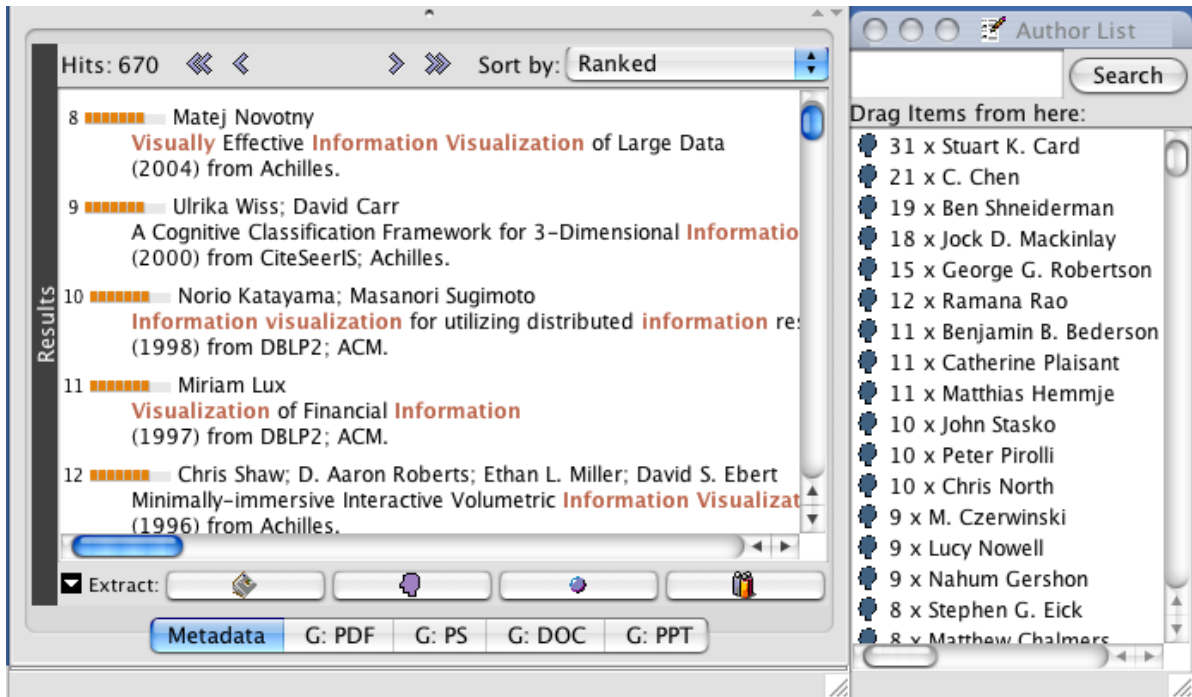


Abbildung 4.10: Extrahierte Autoren einer Resultatliste

Eine gängige Visualisierungsform dafür ist eine Begriffswolke (Tag-Cloud), bei der die Größe und Hervorhebung einzelner Terme die Häufigkeit des Auftretens codieren – so treten häufige Terme visuell hervor (siehe Abbildung 4.11).

- Die Ergebnisse können nach bestimmten Kriterien zu Gruppen zusammengefasst werden (Dynamisches Clustering, siehe Abbildung 4.5.4).

Zu diesem Zweck wurde ein Clustering-Tool integriert, das über mehrere semantische Merkmale Cluster berechnen kann und nach der Recherche auf das Ergebnis angewandt werden kann. Es kann auch direkt als eigenständiges Recherchewerkzeug aufgerufen und eigenständig genutzt werden. Eine Beschreibung des Werkzeugs MLHT-Browser findet sich in Chojnacki u. Fischer (2005).

Mit den hier aufgeführten unterstützenden Maßnahmen werden die Anwender in die Lage versetzt, bessere und fehlerfreiere Anfragen zu spezifizieren. Sie kommen durch die Reformulierungshilfen auch schnell an Terme, die sich dazu eignen, die Anfragen zielgerichtet zu verfeinern. Durch diese Unterstützung wird der Nachteil der langen Anfrageverarbeitungszeiten, der durch die föderierte Anfragestrategie von DAFFODIL unvermeidbar ist, mit einer besseren Strategie gemildert.

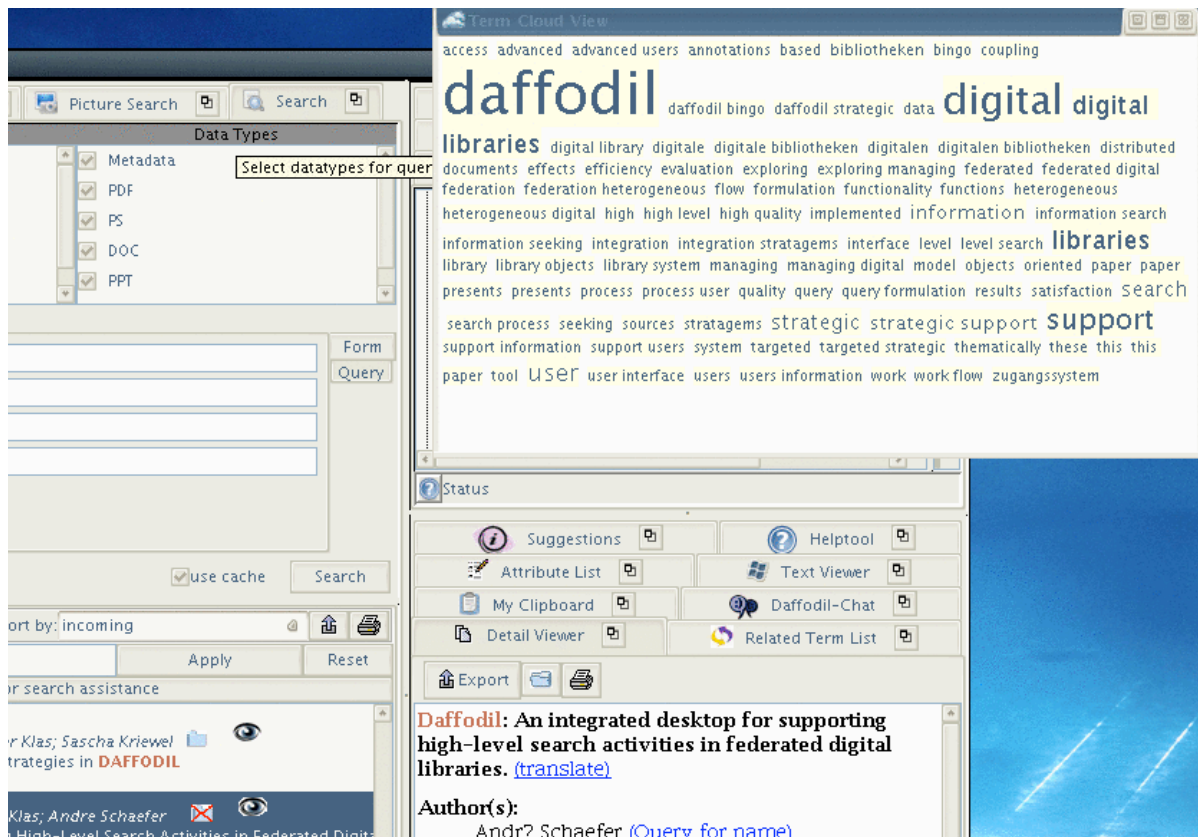


Abbildung 4.11: Extrahierte Terme als Tag-Cloud

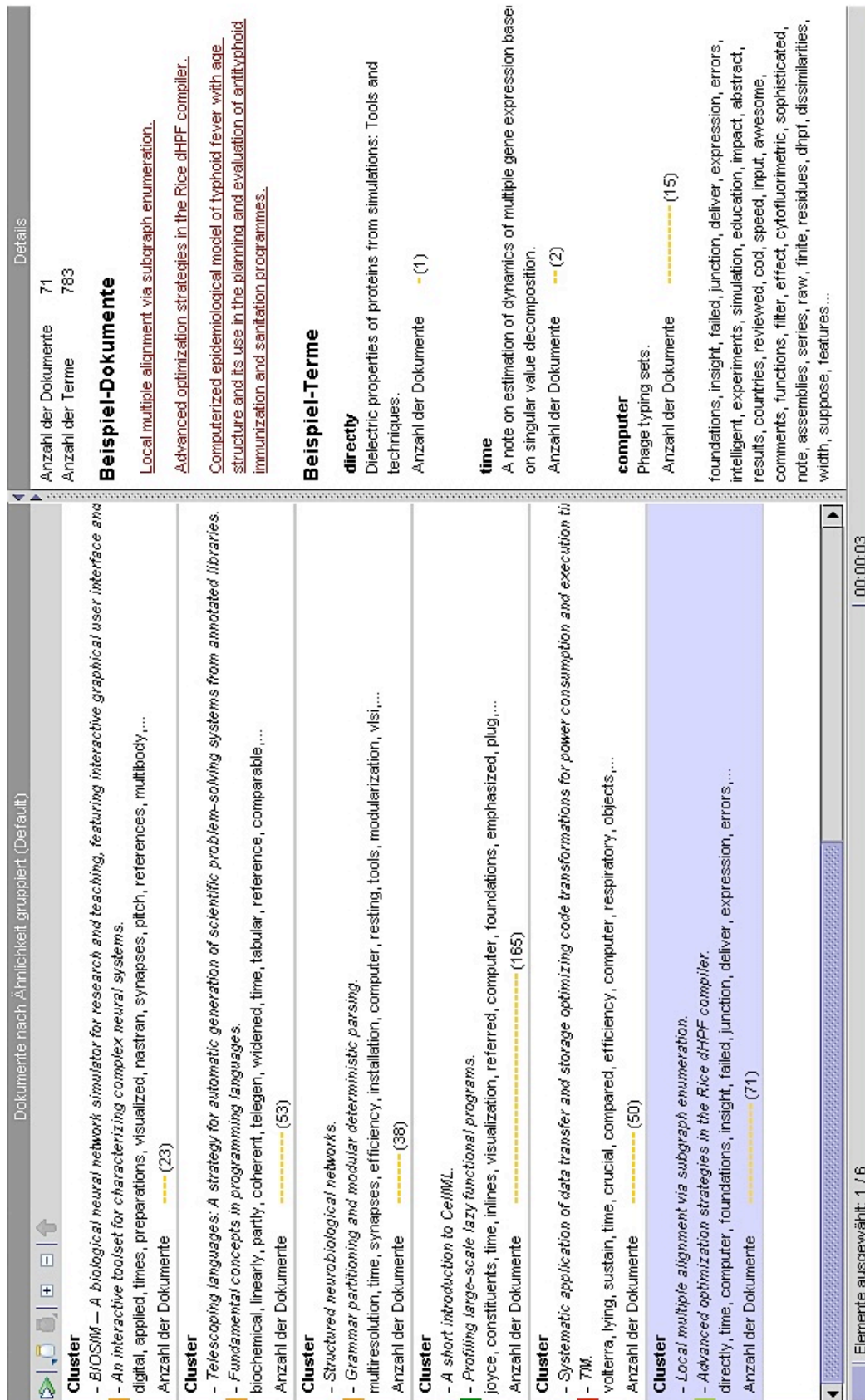


Abbildung 4.12: Clustering der Ergebnisse zu semantischen Gruppen

Die Agenten auf der Service-Ebene und im User Interface Client arbeiten dem Nutzer zu, indem sie die notwendigen Dienste und Funktionen vorausberechnen und zur Verfügung stellen.

4.5.5 Aufbau und Pflege der persönlichen Handbibliothek

Wenn nach einem Rechercheabschnitt einige relevante Ergebnisse gefunden wurden, steht dem Anwender die persönliche Handbibliothek zur langfristigen Ablage und Verwaltung seiner Dokumente zur Verfügung. Da Ablagehandlungen demnach ein Indiz sind, welche Themen den Anwender interessieren, ist die Handbibliothek eine geeignete Quelle für Agentenfunktionen.

Für eine Menge von abgelegten Dokumenten in einem Ordner lassen sich so Empfehlungen (*Recommendations*) berechnen, die inhaltlich passen. So lässt sich eine JITIR-Funktion (vgl. Abschnitt 3.3.1) an eine Benutzeraktion auf einfache Weise koppeln. Der Benutzer legt einige Dokumente oder auch Autor- oder Term-Objekte in die Ablage und aktiviert die Recommendation-Funktion. Diese empfiehlt nun nach asynchroner Berechnung zu einem späteren Zeitpunkt einschlägige Treffer. Hinter dieser Empfehlung können dabei sehr unterschiedliche Methoden der Berechnung angebunden sein. Zum einen können ähnliche Dokumente auf Basis der Retrievalfunktionen einiger Quellen abgerufen werden. Zum anderen können auch Statistiken über die Handbibliotheken vieler Benutzer berechnet werden.

Zudem kann die persönliche Ablage nicht nur dem Recommendation-Agenten freigegeben werden, sondern auch anderen Benutzern, die dann wiederum interessante Dokumente und Metadaten dort ablegen können und so ebenfalls zum kollaborativen, dynamischen Aufbau der persönlichen Handbibliothek beitragen können.

Die Konzepte der Recommendation-Funktionen auf Basis der Handbibliothek sind in der Diplomarbeit von Look (2003) ausgearbeitet und implementiert worden.

4.5.5.1 Awareness

Bei einer dynamisch veränderlichen persönlichen Handbibliothek, in der durch die im vorigen Abschnitt genannten Recommendation-Funktionen oder durch kollaborative Anreicherung stetig neue Information hinzukommt, ist es wichtig, den Anwender auf die neuen Elemente hinzuweisen – er soll sich über die Veränderungen bewusst werden. Solche Hinweise werden *Awareness*-Funktionen genannt.

In der persönlichen Handbibliothek geschieht dies durch grafische Markierungen, nach dem Prinzip der Path-Marker. Die Icons der Objekte in der Ablage werden durch Dekorationen ergänzt, so dass der Anwender sehen kann, welche Items neu sind:



Die Dekorationen werden dabei in der Ordner-Hierarchie der Ablage hochgereicht, so dass man diesem Pfad folgen kann, um zu den markierten Elementen zu gelangen.

Die Awareness-Funktionen in Daffodil werden eingehend in der Diplomarbeit von Roderfeld (2004) diskutiert.

4.5.5.2 Monitoring

Zudem können die Anwender, basierend auf einem Objekt⁴, stehende Anfragen definieren, die regelmäßig ausgewertet werden. Die gefundenen neuen Objekte werden auf Wunsch in die Handbibliothek gelegt, oder dem Anwender wird eine E-Mail mit den Hinweisen zugesandt.

4.5.5.3 Recommendation

Der Inhalt eines Bereiches der Handbibliothek kann auch genutzt werden, um Empfehlungen zu generieren. Wenn andere Nutzer ähnliche Dokumente in ihrem Ablagebereich haben, lassen sich aus deren Interessen Ähnlichkeiten ableiten und so überindividuelle Empfehlungen generieren.

Ebenso lassen sich, wie schon im Allgemeinen diskutiert, Dienste nutzen, die bereits für Internetinhalte relevante Empfehlungen berechnen, wie CiteSeer, Scholar Google, die ACM oder BINGO (vergleiche auch Theobald u. Klas (2004)).

4.6 Benutzungsoberfläche

Die Benutzungsoberfläche, der *Graphical User Interface Client* (GUI Client) ist eine eigenständige Anwendung, die über Nachrichtenaustausch an die verteilten Agenten

⁴etwa ein Autor, Journal oder Konferenz

im Kernsystem gekoppelt ist. Sie dient dazu, einen einfachen und leicht zu installierenden Zugang zum Daffodil-System zu ermöglichen. Voraussetzung ist lediglich ein installiertes Java-Laufzeitsystem.

Im Folgenden wird auf die Konzeption der grafischen Oberfläche einschließlich der wesentlichen Werkzeuge eingegangen.

4.6.1 Anforderungen

Eine Benutzungsoberfläche für das verteilte, agentengestützte Informationssystem DAFFODIL muss einige allgemeine Anforderungen erfüllen, um dem Anspruch gerecht zu werden, ein flexibles Werkzeug zu sein, das den Informationssuchenden in ihrer jeweiligen Arbeitsumgebung stets zur Verfügung steht.

- *Modularität*
Der Umfang an hilfreichen Werkzeugen für die Recherche soll dynamisch wachsen können. Es soll möglich sein, ein neues Werkzeug anzubieten, ohne den Rest der Anwendung verändern oder neu konzipieren zu müssen.
- *Plattformunabhängigkeit*
Der Anwender soll nicht auf eine bestimmte Computerplattform festgelegt sein, sondern das DAFFODIL-System aus seiner gewohnten Arbeitsumgebung heraus nutzen können.
- *Nachrichtenbasierte Architektur*
Die Komponenten in DAFFODIL sollen nur lose gekoppelt sein; das heißt, es sollen keine festen Referenzen und technischen Abhängigkeiten eingebaut werden müssen, wenn zwei Werkzeuge Informationen austauschen sollen. Es soll ebenso die Anbindung zwischen einem Werkzeug und den Informationsdiensten im DAFFODIL-Backend auf dynamisch ausgetauschten Nachrichten beruhen.
- *Einfacher Benutzerzugang*
In der Vorphase zu DAFFODIL wurde zunächst der Zugang mittels einer reinen HTML Web-Applikation getestet. Für die reine Recherche ist dieser Ansatz ausreichend. Um den erweiterten Anforderungen bzgl. flexibler Interaktion, sinnvollen Antwort- und Reaktionszeiten und den besonderen Forderungen des WOB-Modells nach dynamischer Anpassung und kontextsensitiver Durchlässigkeit zu entsprechen, war die Entwicklung einer GUI-Applikation, die beim Anwender lokal läuft, vorzuziehen.

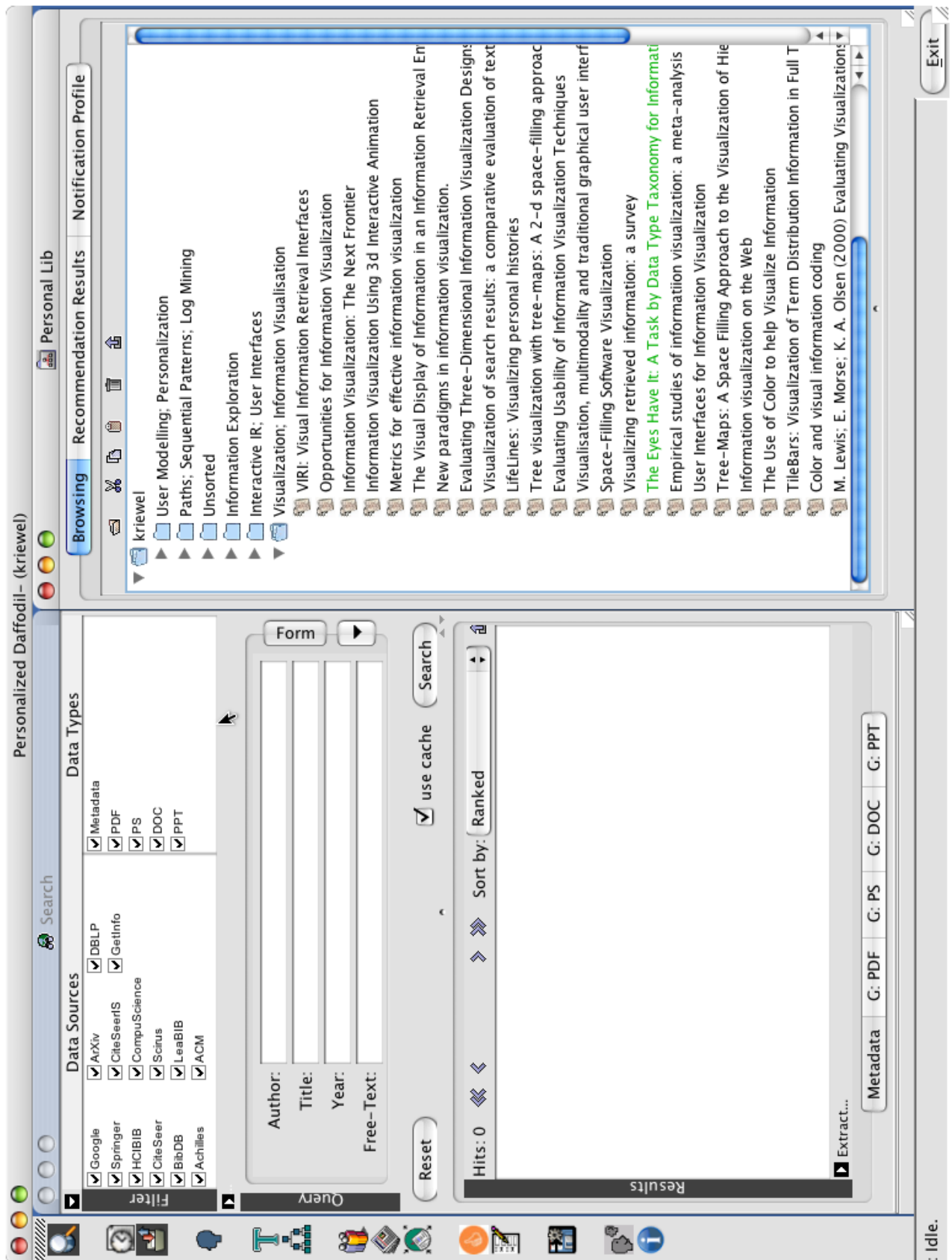


Abbildung 4.13: Daffodil Desktop mit Suchwerkzeug und Handbibliothek

- *Adaptivität*

Das System soll sowohl bezüglich des von Benutzern gewünschten Systemverhaltens als auch bezüglich der jeweils präferierten Inhalte adaptierbar sein und sich auch adaptiv verhalten. Letzteres soll insbesondere für komplexere Aktionsfolgen gelten.

- *Proaktivität*

Das System soll in der Lage sein, von sich aus Vorschläge zu machen oder Aktionen auszuführen. Zu diesem Zweck sollen proaktive, autonome Agenten eingesetzt werden, die durch Beobachtung, etwa der Handbibliothek eines Anwenders, selbständig weiterführende Informationen sammeln und den Anwender anbieten können.

- *Ergonomie*

Die Gestaltung der Benutzungsoberfläche im DAFFODIL-System basiert auf dem WOB-Modell (Vergleiche Abschnitt 4.3); die Konzepte dieses Modells sollen für die Objekte und Funktionen Anwendung finden. Insbesondere sollen die direkte, objektorientierte Manipulation von Informationsobjekten sowie die kontextsensitive Durchlässigkeit die Anwender bei der Arbeit mit verschiedenen Werkzeugen unterstützen.

4.6.2 Graphische Oberfläche und ihre Werkzeuge

Die Auswahl der im weiteren Verlauf vorgestellten Werkzeuge konzentriert sich auf die wesentlichsten Dienste, die den Benutzer dieses Systems mit höheren Suchfunktionalitäten unterstützen.

Der DAFFODIL-Desktop (siehe Abbildung 4.13) ist auf einfache Erweiterbarkeit ausgelegt. Die Integration von weiteren Werkzeugen ist daher leicht durchzuführen.

4.6.2.1 Suche

Das Suchwerkzeug, dargestellt in Abbildung 4.14, stellt einen der möglichen Ausgangspunkte einer Literaturrecherche in DAFFODIL dar. Es bietet eine für den Benutzer einfach zu handhabende Eingabemaske, die es erlaubt, einheitliche Anfragen an die verteilten Bibliotheken zu formulieren und dadurch die Suchdomäne durch Auswahl einiger oder aller digitalen Bibliotheken zu spezifizieren.

The screenshot shows the Daffodil Search application window. It features a 'Filter' section on the left with a vertical scrollbar. The main area is divided into two columns: 'Data Sources' and 'Data Types'. Below these is a 'Form' section with input fields for 'Author:', 'Title:', 'Year:', and 'Free-Text:'. The 'Title:' field contains the text 'internet; library'. Below the form is a 'Query' section showing the generated query: 'Title=internet AND Title=library'. At the bottom, there are buttons for 'Clear', 'Forward', 'Search', and a checkbox for 'use cache' which is checked.

Data Sources		Data Types
<input checked="" type="checkbox"/> ACM	<input checked="" type="checkbox"/> Scirus	<input checked="" type="checkbox"/> Metadata
<input checked="" type="checkbox"/> PLib	<input checked="" type="checkbox"/> CiteSeer	<input checked="" type="checkbox"/> PDF
<input checked="" type="checkbox"/> GetInfo	<input checked="" type="checkbox"/> Google	<input checked="" type="checkbox"/> PS
<input checked="" type="checkbox"/> LeaBIB	<input checked="" type="checkbox"/> DBLP	<input checked="" type="checkbox"/> DOC
<input checked="" type="checkbox"/> CompuScience	<input checked="" type="checkbox"/> HCIBIB	<input checked="" type="checkbox"/> PPT
<input checked="" type="checkbox"/> ArXiv	<input checked="" type="checkbox"/> Achilles	
<input checked="" type="checkbox"/> Springer		
<input checked="" type="checkbox"/> BibDB		

Form fields:

Author:

Title:

Year:

Free-Text:

Use form fields to specify query. The ; splits two query clauses.
 The default connection operator is **AND**.
 The query can use **OR, AND, NOT, ()** - (edit below or on result display)

Form

Query: Title=internet AND Title=library

Clear ☒ use cache

Abbildung 4.14: Suchwerkzeug

Die Anfragen des Suchwerkzeuges werden über eine föderierte Suche an die Informationsanbieter weitergeleitet. Zur Erfassung der Anbieterdaten werden Wrapper eingesetzt. Diese nutzen hierzu in der Regel das WWW-Suchformular der entsprechenden digitalen Bibliothek. Diese wird durch das DAFFODIL-Wrappertoolkit angebunden.

Die Anfragen werden parallel bearbeitet, die Ergebnisse anschließend zusammengeführt und dem Benutzer auf homogene Weise zur Betrachtung präsentiert. Dabei werden über unaufdringliche Icons (siehe Abbildung 4.15) bereits betrachtete, abgespeicherte oder anderweitig bearbeitete Dokumente kenntlich gemacht.

Da die Relevanzgewichtung bei einer föderierten Suche nicht mit einfachen Mitteln normalisiert werden kann, wird bei der Integration der Ergebnisse nachträglich gewichtet. Das Ranking geschieht mittels des Retrievalsystems PIRE (siehe (Nottelmann 2005)) auf Basis von BM-25-Indexierungsgewichten.

Mit Hilfe einer Extraktionsfunktion kann dem Benutzer ein aus der Resultatliste gewonnener Überblick über die zentralen Autoren, Konferenzen, Journale oder Begrifflichkeiten eines Suchergebnisses geboten werden. Dadurch wird z.B. das Auffinden von Hauptakteuren eines Themengebietes erleichtert und zudem stehen dem Benutzer neue Ansatzpunkte für die weitere Suche zur Verfügung.

4.6.2.2 Persönliche Handbibliothek

In der persönlichen Handbibliothek (siehe Abbildung 4.16) ist das Speichern von Ergebnissen, Volltexten, Stichworten oder Suchtermen, Autoren, Konferenzen oder Journalen über den Kontext einer Suche hinaus möglich. In persönlichen Ordnern können Benutzer Ergebnisse strukturiert ablegen und so über mehrere Suchsitzungen hinweg ein Archiv ihrer persönlichen Literaturrecherchen langfristig aufbauen. Für die Benutzung der Resultate außerhalb des Systems stehen Exportfunktionalitäten zur Verfügung. In so genannten Gruppenordnern besteht ebenfalls die Möglichkeit zur Annotation an Objekten, einschließlich bestehender Annotationen, wodurch das Konzept der kollaborativen Recherche unterstützt wird. Auf Grundlage der persönlichen Ordner bietet das System Awareness und Empfehlungen.

Um ein Thema über einen längeren Zeitraum bearbeiten und verfolgen zu können, kann man derzeit die Objekte Autor, Suchanfrage, Journal, Konferenz und Webseite mit diesem Konzept beobachten und sich durch einen Notification Service über neue

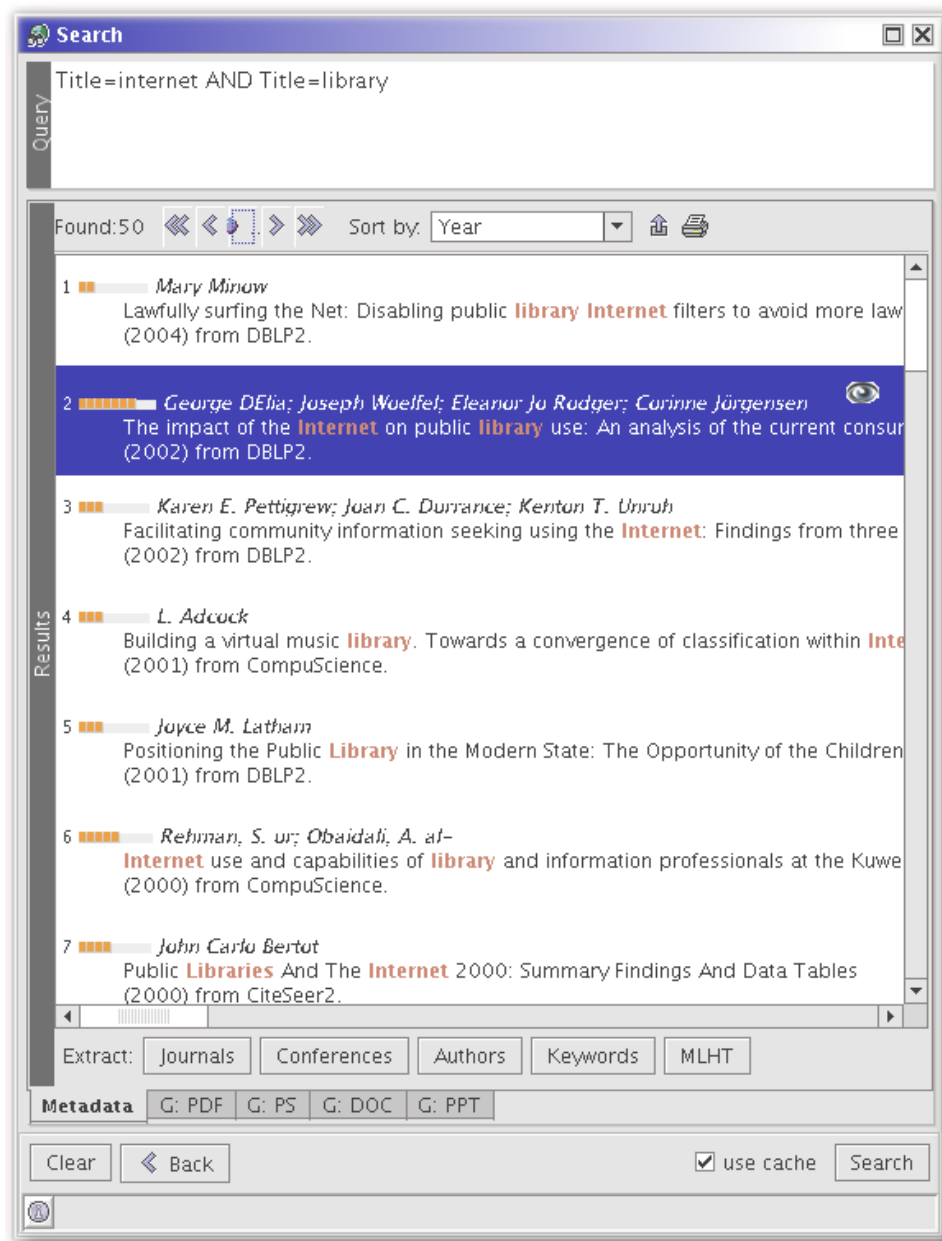


Abbildung 4.15: Gewichtete Resultatliste mit Icons

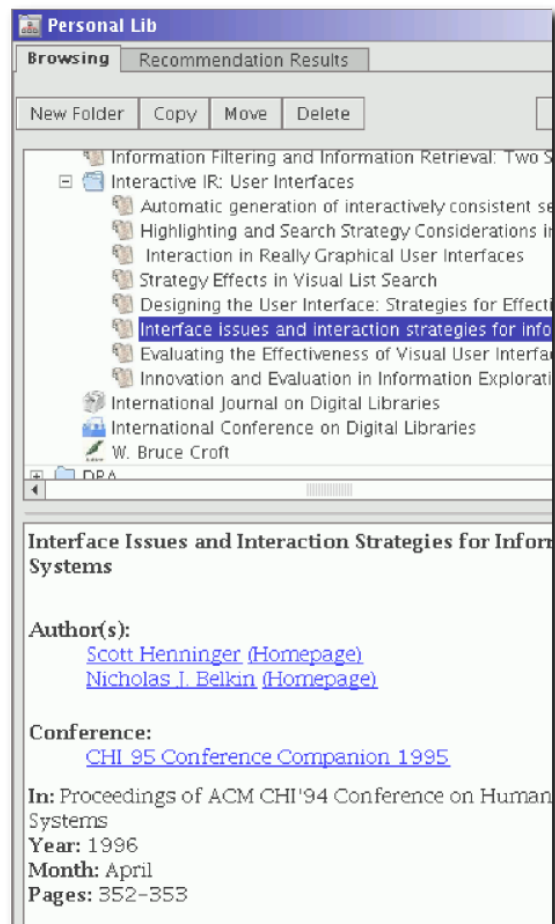


Abbildung 4.16: Persönliche Handbibliothek

Resultate informieren lassen. Dies beinhaltet eine automatische Suche nach neuer Information in festgelegten Intervallen. Die so neu gefundenen Ergebnisse werden in die persönliche Handbibliothek eingestellt und kenntlich gemacht, so dass der Benutzer sie als neue Informationseinträge erkennt.

4.6.2.3 Referenzen und Zitationen

Ausgehend von einem bekannten Dokument, etwa aus der Resultatliste oder der persönlichen Handbibliothek, erlaubt das Referenzwerkzeug die Suche nach anderen Objekten. Wenn es sich hierbei etwa um Dokumente handelt, können diese zitiert worden sein bzw. das ursprüngliche Dokument selbst zitieren. Das Werkzeug kann durch Drag and Drop eines Dokuments aus anderen Werkzeugen heraus aktiviert werden. Die Ergebnisse lassen sich direkt weiterverwenden und können wiederum in der persönlichen Handbibliothek abgespeichert werden.

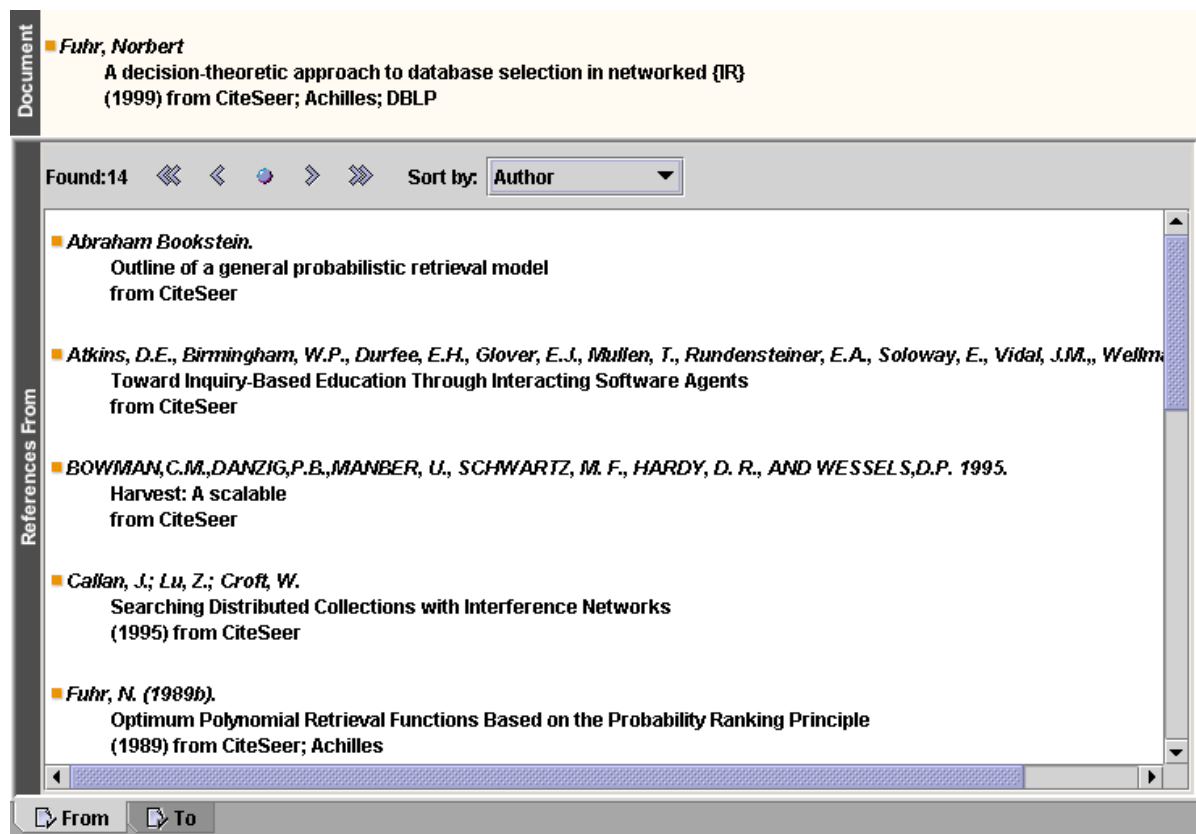


Abbildung 4.17: Referenzen und Zitationen

4.6.2.4 Autorennetzwerke

Ein weiteres oft benutztes und von DAFFODIL unterstütztes Suchstrategem ist die Autorensuche. Ausgehend von einem Autor, dessen Relevanz für das Suchinteresse bekannt ist, kann im DAFFODIL-System nach weiteren Publikationen dieses Autors gesucht oder die Koautorenbeziehungen des Autors für den Aufbau eines Beziehungsnetzwerks genutzt werden.

In einem solchen Koautorennetzwerk, das auch als Beziehungsgraph visualisiert werden kann (siehe Abbildung 4.18), lassen sich leicht sowohl zentrale Autoren erkennen⁵, als auch Autoren finden, die häufig gemeinsam publizieren.

Darüberhinaus ist es für die Recherche oft wichtig, die zentralen Personen, die sich mit dem relevanten Thema befassen, zu identifizieren (Experten-Suche): Ausgehend von einem als relevant identifizierten Dokument von einem weniger zentralen Verfasser lassen sich über die Soziale-Netzwerk-Analyse wichtige Personen in dessen Publikationsumfeld identifizieren.

Zudem kann das so berechnete Netzwerk herangezogen werden, um die Ergebnisse einer Recherche in eine relevanzbezogene Reihenfolge zu bringen (Ranking). Empirische Untersuchungen belegen, dass es eine starke statistische Korrelation gibt, zwischen der Zentralität der Autoren einer Veröffentlichung und der Relevanzbewertung ihrer Veröffentlichung. Einerseits kann im Rahmen einer föderierten Suche oft nicht einheitlich mit einem Verfahren nach Inverse Document Frequency (IDF) gerankt werden. Andererseits führen die über die soziale Netzwerkanalyse berechneten Rankings nachweislich zu einer Erhöhung der Präzision unter den Top 20 Treffern. Hierbei werden auch andere Dokumente präferiert als bei einem IDF Verfahren, somit ist es sinnvoll, diese Ranking-Methode als zusätzliches Sortierkriterium anzubieten. Dieses Verfahren wurde in Daffodil realisiert und kann neben einem Post-Retrieval-IDF-Ansatz zur Sortierung der Ergebnisliste gewählt werden⁶.

Das Konzept zur Berechnung und Darstellung der Autorennetze ist bei Mutschke (2001) und Mutschke (2004) beschrieben.

Generell zur Verwendung von Graphen in direktmanipulativen Oberflächen vergleiche auch Abschnitt 2.5.5.

⁵Das ist dann die Expertensuche, vergleiche Mutschke (2004, Abschnitt 5.1.2, S. 37 ff.)

⁶Vergleiche hierzu Mutschke (2004, S. 43 ff.)

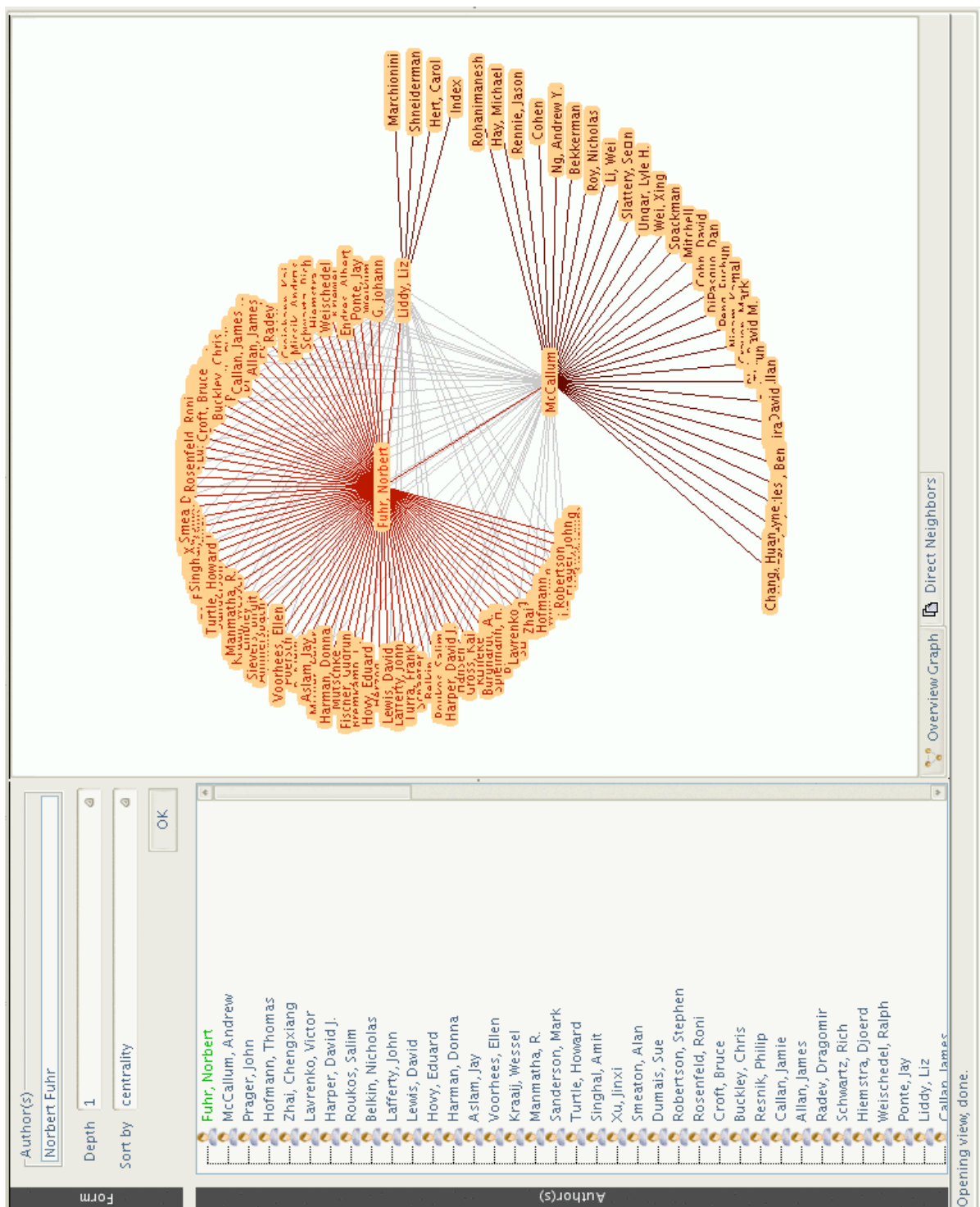


Abbildung 4.18: Koautoren-Netz Analyse Werkzeug

4.6.2.5 Journale und Konferenzen

Zum Blättern und Suchen in Zeitschriftenjahrgängen bzw. Konferenzbänden steht das Journal- und Konferenzwerkzeug (siehe Abbildung 4.19) zur Verfügung. Hier kann nach Titeln von wissenschaftlichen Journalen oder Konferenzen gesucht werden, um innerhalb der Ergebnisse zu browsen. Hierzu besteht oft der direkte Zugriff auf Metadaten oder Volltextlinks.

Der Einsatz der Werkzeuge kann Ausgangspunkt oder Zwischenschritt eines umfangreichen Suchplans sein. Explizite Verknüpfungen in den Detailansichten von Suchergebnissen weisen auf eine Zeitschrift oder einen Konferenzband hin, in dem ein Dokument veröffentlicht worden ist, und springen bei Aktivierung direkt in das entsprechende Werkzeug.

4.7 Fazit Daffodil

Mit dem Projekt DAFFODIL steht eine flexible Infrastruktur zum Bauen ganzheitlicher Informationssysteme zur Verfügung. Die Möglichkeit der föderierten Anbindung unterschiedlicher heterogener Informationsquellen zusammen mit einer reichhaltigen Dienstestruktur und der Flexibilität und Anpassungsfähigkeit der Software-Agenten macht eine Umsetzung moderner Konzepte zur Unterstützung der Informationskompetenz und der strategischen Unterstützung der Anwender möglich. Das erstreckt sich auch auf die Arbeitsphasen nach der Recherche und hilft beim Verwalten und Integrieren der Information bis zur Erstellung neuer Veröffentlichungen.

Die auf dieser Basis umgesetzten proaktiven Komponenten und Dienste können zur Evaluation verschiedener Konzepte und ihrer Auswirkungen auf Retrievalqualität und Softwareergonomie dienen.

Die Evaluationen, die die Projektentwicklung begleitet haben, und deren Ergebnisse werden im nächsten Kapitel dargestellt.

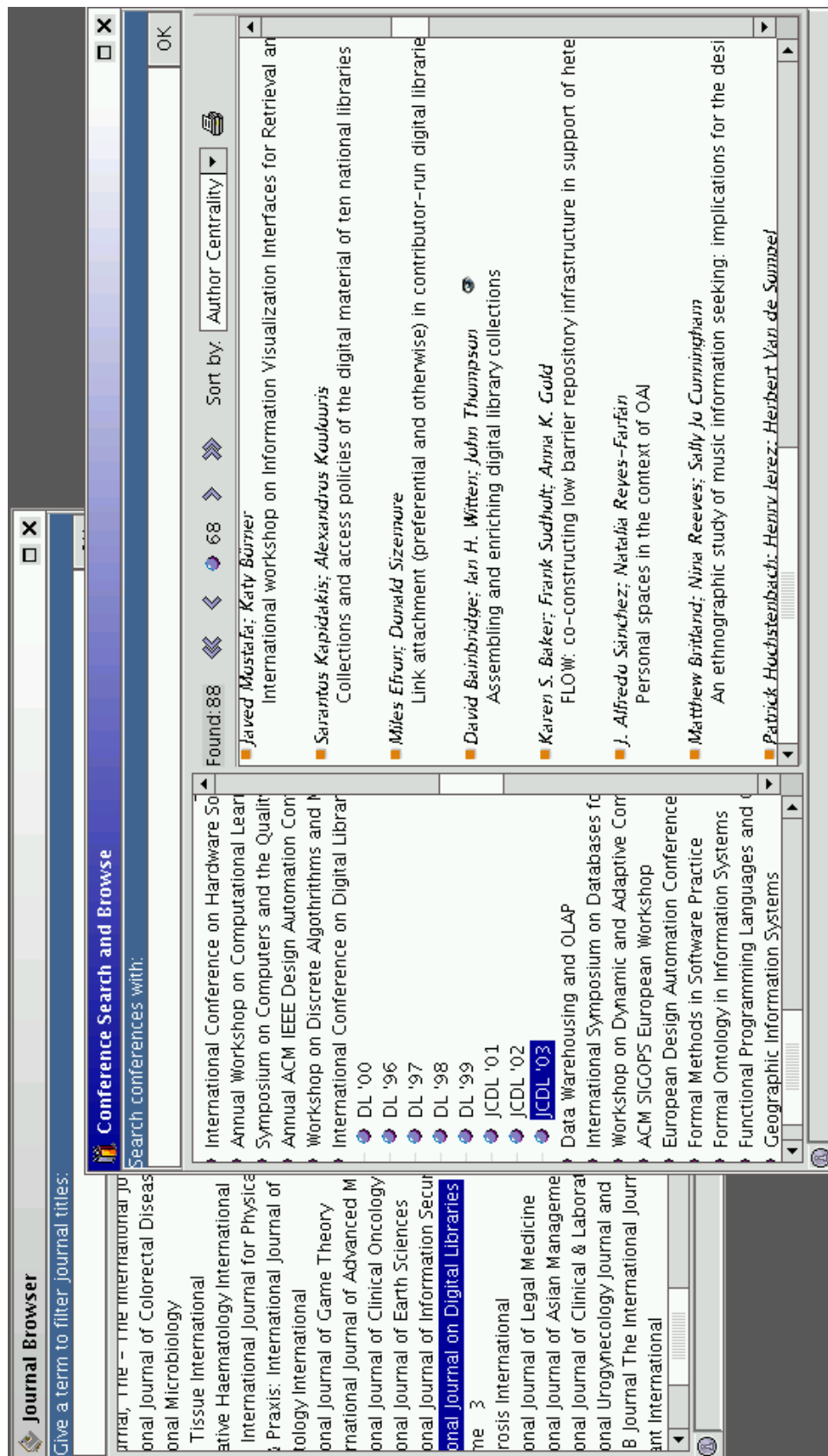


Abbildung 4.19: Journal- und Konferenzbrowser

5 Evaluation

Das Projekt DAFFODIL wurde durchgängig von empirischen Evaluationen begleitet. Im Folgenden werden diese Studien beschrieben und jeweils die für die Thesen dieser Arbeit relevanten Ergebnisse diskutiert.

In den Tests wurden zum Abschluss der ersten und zweiten Projekt-Förderphase die Akzeptanz und Qualität der jeweils umgesetzten Konzepte in qualitativen Tests mit Nutzern aus der Zielgruppe getestet.

Quantitative Tests zur Retrievalqualität wie sie in Eibl (2000) anhand der unabhängig relevanzbewerteten GIRT-Daten durchgeführt wurden, hängen von einer entsprechenden, zur Domäne passenden, Datenbasis ab. Analoge Tests konnten mit DAFFODIL aufgrund der heterogenen und dynamischen Datenbasis von DAFFODIL nicht durchgeführt werden.

Es wurden jedoch in Klas u. a. (2007) Methoden erarbeitet, um DAFFODIL als experimentelles Framework zur Evaluation bestehender Informationssysteme einzusetzen. Insbesondere wurde ein Logging-Framework definiert und eingeführt, das ausgehend von einer idealtypischen Konzeption einer digitalen Bibliothek die ganzheitliche Erfassung der charakteristischen Verhaltensweisen (*Umgang*) eines Anwenders gegenüber einer Datenbasis erfassen soll.

Der Akzeptanztest in Phase I beruht auf den Empfehlungen von Nielsen (1994) zur *Heuristischen Evaluation*. Die Evaluation nach Phase II verbindet diesen Ansatz mit dem der *realistischen Arbeitsumgebung* und der *simulierten Arbeitssituation* nach Borlund (2003).

Exkurs: Heuristische Evaluation Eine von Nielsen (1994) vorgestellte Methode nennt sich *Heuristische Evaluation*. Diese Methode beruht auf einer Sequenz von Einzelinterviews. Den Teilnehmern wird zunächst kurz die Software vorgestellt. Daraufhin sollen sie selbsttätig die Software benutzen und langsam die Möglichkeiten

explorieren. Zusätzlich wird den Personen eine Liste mit Heuristiken an die Hand gegeben, die sie als Bewertungsgrundlage bei auftretenden Problemen heranziehen sollen. Diese Heuristiken sollen die bewusste Wahrnehmung unterstützen, sowie eine gemeinsame Basis der Bewertungen herstellen, um eine gewisse Objektivität beizubehalten. Der Beobachter protokolliert den Verlauf der Sitzung mit.

Ziel der Evaluation ist es, eine Liste mit Usability-Problemen zu erstellen, wobei jedes dieser Probleme anhand der Heuristiken begründet sein soll. Ergebnis des Verfahrens, nach dem es mit verschiedenen Personen etwa dreimal bis fünfmal wiederholt worden ist, ist eine Aufstellung von Benutzbarkeitsproblemen mit Hinweisen auf deren Ursache. Nach Nielsen lassen sich so auf einfache Weise etwa 80% der offensichtlichen Probleme herausfinden. Aufgrund der einfachen Durchführung bei gleichzeitiger hoher Effizienz wurde dieses Verfahren für die qualitativen Benutzer-tests ausgewählt.

Nielsen schlägt die folgende Liste von heuristischen Kriterien¹ vor, um eine Software auf ihre Gebrauchstauglichkeit zu prüfen:

Visibility of system status The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

¹http://www.useit.com/papers/heuristic/heuristic_list.html

Recognition rather than recall Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

5.1 Daffodil Phase I

Zum Abschluss der ersten Projektphase 2002 wurde der damalige Prototyp samt Dienste-Infrastruktur mittels eines Akzeptanz-Tests evaluiert. Eine detaillierte Beschreibung findet sich auch in Fuhr u. a. (2002).

5.1.1 Ziele

Ziel der Evaluation war es, den ersten funktionsfähigen Prototypen von DAFFODIL mit Vertretern der Zielgruppe zu testen. Die Funktionsfähigkeit sollte sichergestellt werden, und das Konzept der Verteilung der Funktionalität auf verschiedene Werkzeuge sollte auf seine Akzeptanz geprüft werden.

5.1.2 Umfang und Teilnehmer

DAFFODIL richtet sich an Studenten, Wissenschaftler und Bibliothekare aus dem Bereich der Informatik. Es soll ein Recherchewerkzeug für diese Nutzergruppen sein und deren Anforderungen möglichst gut abdecken.

An diesem ersten Test nahmen daher 14 Informatikstudenten der Universität Dortmund in teil. Jeder Teilnehmer nutzte die Software einzeln. Die einzelnen Sitzungen dauerten ungefähr eine Zeitstunde. Bei der Bearbeitung der Aufgaben war ein Protokollant zugegen, der die Äußerungen der Teilnehmer und eigene Beobachtungen protokollierte.

Eine Sitzung bestand aus drei Abschnitten; die ersten beiden Abschnitte wurden mittels Video-Protokoll dokumentiert.

- Die Teilnehmer sollten Daffodil installieren.
- Zunächst sollten sie einige leichtere Aufgaben sowie eine komplexere Recherche durchführen und dabei nach Möglichkeit laut kommentieren, was sie tun und was sie stört.
- Anschließend wurde jeder Teilnehmer gebeten, einen Fragebogen zu beantworten, der auf den heuristischen Kriterien von Nielsen beruhte (siehe oben) und den Teilnehmer bat, Rückmeldungen zu geben, falls die Software in einem der von den Heuristiken benannten Bereiche Probleme aufweist.

Die Evaluation belegte Probleme, die durch die semantische Abbildung der Anfragen auf die heterogenen Datenquellen entstanden. Ermittelt wurde auch der Bedarf an einer persönlichen Handbibliothek und proaktiver Hilfe bei nicht zufriedenstellenden Antworten, wie der Vorlage von Termen zur Anfrageerweiterung oder der Anfragekorrektur.

Die Beobachtungen bei den ersten Tests belegen, dass die Nutzer, wenn sie mit einer konkreten Retrievalaufgabe betraut sind, zunächst unsicher sind, wie und wo mit der Recherche zu beginnen ist. Die Arbeitsumgebung von DAFFODIL konnte den Zugang zu den diversen Informationsquellen im Bereich der Informatik erleichtern – die Probanden hätten ohne die Infrastruktur die meisten der Quellen ignoriert und sich auf ihre jeweiligen OPAC- und Google-Kenntnisse gestützt. In sofern standen sie der Reichhaltigkeit des Informationsangebotes positiv gegenüber.

Auch die höheren Suchfunktionen, insbesondere in Form der Koautorensuche, wurde sehr positiv beurteilt – diese unbekannte Funktion war jedoch erklärungsbedürftig. Die strategische Idee, sich das soziale Umfeld zu einem einschlägigen Autoren zu erschließen, war den meisten Probanden neu.

Es zeigte sich, dass das zeitliche Verhalten in einem verteilten und komplexen Informationssystem ein sehr kritischer Punkt ist, und dass Last- und Performancetests zu jedem Entwicklungszeitpunkt notwendig sind. Erst unterhalb einer kritischen Antwortzeitschwelle von maximal 90 Sekunden gab es überhaupt genug Akzeptanz um ein Suchergebnis abzuwarten. Dass die Probanden überhaupt so lange zu warten bereit waren, erklärt sich aus den Vorteilen, die sie sich durch die breit angelegte föderierte Recherche versprachen.

Auf Basis der Testresultate wurde das Interface von Daffodil überarbeitet und verbessert. Die verbesserte Version wurde Grundlage der Arbeiten in Phase II.

5.2 Daffodil Phase II

Am Ende der zweiten Projektphase wurde in einer qualitativen Evaluation (Klas u. a. 2004) untersucht, ob die inzwischen implementierten Maßnahmen zur strategischen Unterstützung den Information-Retrieval-Prozess für den Anwender verbessern können.

Zudem wurden in einer weiteren qualitativen Evaluation (Schaefer u. a. 2005) die proaktiven Vorlagen bei der Anfrageformulierung bewertet.

5.2.1 Strategische Unterstützung

Zum Projektabschluss der zweiten Phase des DAFFODIL Projektes war es bedeutsam, das Hauptziel des Projekabschnitts, nämlich die *strategische Unterstützung*, zu bewerten. In diesem Abschnitt werden Durchführung und Ergebnisse dieser Studie wiedergegeben.

5.2.1.1 Ziele der Evaluation

Bei der Evaluation zur strategischen Unterstützung stand die Sicht des individuellen Anwenders im Vordergrund. Auch stand die Evaluation der vertikalen Funktions-

weise bis hinunter zu den verteilten Diensten vor den Erkenntnissen über die Qualitäten der Benutzerschnittstelle.

Die zu evaluierenden Thesen waren:

- a) Durch die enge Integration der Informationsquellen und die strategische Unterstützung durch höhere Suchfunktionen wird es für die Benutzer einfacher, die Ziele der Recherche zu erreichen.
- b) Die Rechercheziele werden schneller erreicht, als bei der nicht-integrierten Nutzung verschiedener Informationsquellen und allgemeiner Suchmaschinen.

Proaktive Funktionen heben die Stufe der Systemunterstützung. Die Arbeitsbelastung sollte also sinken, während die kognitive Arbeit nicht durch ablenkende Mechanismen gestört wird. Somit war zu erwarten, dass Effizienz und Effektivität steigen.

In die Bewertung flossen dabei nur der Zugriff auf bibliographische Informationen ein. DAFFODIL ist selbst kein Informationsangebot, sondern nur eine Infrastruktur zur Informationsvermittlung. Aus diesem Grund wurden auch Kriterien wie Persistenz und ökonomische Aspekte außer Acht gelassen.

Die Bewertung wurde anhand der Kriterien der Relevanz der Ergebnisse, der Qualität und der Benutzerzufriedenheit vorgenommen.

5.2.1.2 Durchführung der Evaluation und Resultate

Die Evaluation wurde an zwei separaten Nutzergruppen mit unterschiedlichen Methoden durchgeführt. Die Benutzer wurden vorab anhand ihrer Kenntnisse kategorisiert:

Gruppe A Die Gruppe der Nicht-Experten bekam eine Serie an kontrollierten Aufgaben und einen Evaluationsfragebogen. Diese Gruppe setzte sich aus 14 Informatikstudenten und wissenschaftlichen Mitarbeitern aus dem Bereich der Informatik zusammen.

Gruppe B Als Experten galten Professoren und Bibliothekare. Experten bekamen zur Aufgabe gestellt, ein Thema aus ihrem aktuellen Interessengebiet zu recherchieren und durften das System nach kurzer Einführung frei benutzen. Die Gruppe der Experten bestand aus je drei Professoren und drei hauptberuflichen Bibliothekaren.

Durchlauf in der Gruppe A Der Gruppe A wurden fünf Aufgaben gestellt:

1. *Known Item (a)*

Es sollte ein Artikel anhand eines gegebenen vollständigen Titels ausfindig gemacht werden. Da der Titel präzise angegeben war, sollte eine Suchmaschine wie Google in der Lage sein, den Artikel schnell ausfindig zu machen.

2. *Known Item (b)*

Wiederum soll ein Artikel anhand eines Titels gefunden werden. Dieses Mal war der Titel aber nur inhaltlich wiedergegeben und nicht mit dem Original identisch. Da es sich um unvollständiges Wissen handelte, war die Rechercheaufgabe etwas komplexer.

3. *All Articles*

Die dritte Aufgabe war, alle Artikel eines gegebenen Autors auf einem ebenfalls gegebenen Arbeitsgebiet zu ermitteln.

4. *Author Interest*

Es sollte herausgefunden werden, welchen Hauptinteressensgebiet ein namentlich genannter Autor verfolgt.

5. *Subject Search*

Es sollte möglichst umfangreiche Information zu einem dem Suchenden gänzlich unbekannten Gebiet gesammelt werden.

Die Auswertung wurde anhand der Fragebögen, aber ebenso gestützt durch Protokolle und die durch die Anwendung aufgezeichneten Log-Dateien, vorgenommen. Zudem wurden die Probanden aufgefordert, als besonders relevant empfundene Daten in der persönlichen Handbibliothek zu sammeln. Die so abgelegten Daten wurden zur Bewertung der Relevanz der Ergebnisse herangezogen.

Die Auswertung erbrachte die folgenden interessanten Ergebnisse:

1. Im einfachsten Fall (Known Item(a)) war die generische Suchmaschine durch ihre Performance im Vorteil. Daffodil brauchte für die föderierte Suche in 10 Quellen ca. 30 Sekunden, die generische Suchmaschine antwortete umgehend. Die Probanden verbrachten jedoch insgesamt nicht weniger Zeit bei der Recherche. Offenbar gleicht die Zeit, die in Anspruch genommen wurde, um das gesuchte Dokument in der Ergebnisliste zu identifizieren, den zeitlichen Aufwand bei der föderierten Suche aus.

2. Wenn der Titel in der angegebenen Form nicht sofort das gewünschte Dokument in der Suchmaschine zu Tage fördert, braucht die Recherche im Schnitt weniger Zeit in DAFFODIL (durchschnittlich 54 Sekunden) als in der generischen Suchmaschine (durchschnittlich 113 Sekunden).
3. Bei der *All Articles*-Aufgabe fanden die Probanden in DAFFODIL durchschnittlich 12 von den relevanten 19 Dokumenten.

Mit der Suchmaschine brachen 12 der 14 Probanden die Suche vorzeitig ab, die beiden übrigen fanden wesentlich mehr als die gefragten Dokumente, waren jedoch nicht in der Lage, die Ergebnisliste von den nicht relevanten Ergebnissen zu bereinigen. Der angegebene Grund zum Abbruch war in den meisten Fällen Frustration über den hohen Zeitaufwand.

Bei dieser Aufgabe war das Informationssystem DAFFODIL klar im Vorteil.

4. Die Ergebnisse zu den komplexeren Aufgaben *Author Interest* und *Subject Search* waren quantitativ weniger aussagekräftig, jedoch wurden die Aufgaben im Schnitt mit hoher Benutzerzufriedenheit gelöst. Die Werkzeuge zur Extraktion von Metadaten für die Anfragereformulierung erhielten positives Feedback.

Zusammenfassend kann gesagt werden, dass die Probanden konzentriert ihre Aufgabe verfolgen konnten. Sie begannen nach der Einarbeitungsphase, dem Werkzeug zu trauen und haben DAFFODIL auch genutzt, um Ergebnisse der WWW-Suchmaschine auf Korrektheit zu prüfen, was zeigt, dass das Vertrauen in die Daffodil Ergebnisse sehr hoch war. Zusammen mit den positiven Urteilen aus den Fragebögen ergibt sich ein hohes Maß an Nutzerzufriedenheit.

Durchlauf in der Gruppe B Die Gruppe von 6 Experten (Professoren und Bibliothekare) sollten einen Analyse der Recherchemöglichkeiten von Daffodil vornehmen. Es wurde angenommen, dass diese Personen über ein hohes Maß an Informationskompetenz verfügen und in ihrer täglichen Arbeit Suchtaktiken und Strategien bereits anwenden.

Die Benutzung des Recherchesystems war thematisch nicht eingeschränkt und die Arbeitsabläufe konnten frei gewählt werden. Das Recherchethema war während der Sitzung fix, wurde von den Probanden zu Beginn der Sitzung jedoch frei gewählt. Die Einführung in das System war die gleiche wie bei Gruppe A, eine circa halbstündige Erläuterung der Werkzeuge und der grundlegenden Funktionsweisen.

Die Auswertung wurde durch Videoaufzeichnung vorgenommen, wobei die Probanden laut denkend vorgehen sollten. Die Einzelsitzungen dauerten zwischen 30 Minuten und zwei Zeitstunden. Die Probanden nutzten das System intuitiv und verfolgten konzentriert ihre Aufgabe.

Die Probanden beurteilten positiv, dass es so reichhaltige Recherchemöglichkeiten gibt und dass die meisten Taktiken, die sie anwenden wollten, auch angeboten und unterstützt wurden. Vermisst wurde, dass es nicht möglich war, bereits erhaltene Resultatlisten miteinander zu kombinieren (etwa eine Schnittmenge zu bilden). Eine Verschlagwortungsmöglichkeit und eine daraus resultierende Kategorisierung wurde ebenfalls vermisst.

Insgesamt wurde DAFFODIL von den Experten ebenfalls sehr positiv beurteilt und als sinnvolle Ergänzung zu den schon verwendeten Recherchewerkzeugen betrachtet. Es wurde auch geäußert, dass es ein Experten-System sei, da sich die Möglichkeiten nicht auf die Anfragen beschränken, sondern dem Nutzer auch Wissen über Suchtaktiken vermittelt.

Somit wurden die erhofften Ergebnisse von Projektphase II, nämlich dass die angebotenen höheren Suchfunktionen und die taktische und strategische Unterstützung zu effektiverem Information Retrieval und höherer Benutzerzufriedenheit führt, durch die Evaluation bestätigt.

Auf die Fragestellung der Ergonomie der Arbeitsteilung zwischen Agenten und Anwender hin betrachtet, zeigt sich, dass die Anwender bei steigender Komplexität der Rechercheziele auch zunehmend echten Bedarf an taktischer und strategische Unterstützung haben. Ohne die Unterstützung, nur mit einer WWW-Suchmaschine ausgestattet, zeigte sich insbesondere bei Gruppe A die Tendenz, den Recherchevorgang frustriert abzubrechen, was bei der Nutzung von DAFFODIL nicht geschah.

Die durch die Agenten erzielten Mehrwertfunktionen und deren integrative Leistung über mehrere heterogene Quellen hinweg wird zunehmend zur Notwendigkeit. Dies gilt ins besondere für Benutzergruppen mit mittlerer oder niedriger Informationskompetenz. Aber auch Experten begrüßen die Integration und die Arbeitserleichterung, die sich daraus ergibt.

5.2.2 Evaluation der proaktiven Vorlage bei der Anfrageformulierung

Das Design der proaktiven Vorlagefunktionen (siehe Abschnitt 4.5.3) wurde von drei qualitativen Evaluationen begleitet:

1. Eine heuristische Evaluation nach Nielsen (1994) mit grafischen Design-Prototypen, bestehend aus Screenshots.
2. Kontrollierte Benutzertests mit zwei Gruppen von Studenten, mit und ohne proaktive Funktionen.
3. Einzel-Benutzertests mit Videoprotokollierung.

5.2.2.1 Ziele der Evaluation

Durch die Evaluation sollten eine Reihe von Erwartungen überprüft werden. Die grundlegende Erwartung ist die Akzeptanz der Funktionen. Diese sollte bereits durch die schon in Abschnitt 4.5.3 beschriebene Vorlage von Screenshots mittels heuristischer Evaluation abgesichert werden.

Die gewählten Parameter für das Timing waren zu prüfen, da das zeitliche Verhalten bei proaktiven Vorlagen ein kritisches Gestaltungsmerkmal ist. Da dieses Verhalten erst mit dem funktionalen Prototyp möglich ist, konnte es erst in den beiden späteren Benutzertests geprüft werden. Ob die Version mit proaktiver Vorlage besser angenommen würde als die Basisversion, sollte im zweiten Test herausgefunden werden.

Nach dem Kennenlernen und Verstehen der Funktionen sollten die Anwender die Funktionen selbstständig weiter intuitiv nutzen. Dies sollte vor allem im dritten Test beobachtet werden.

5.2.2.2 Durchführung und Resultate

Im Folgenden sind die jeweilige Durchführung der Tests zusammen mit den jeweiligen Resultaten beschrieben. Genauere Angaben zu den Auswertungen finden sich in Jordan (2005), die folgenden Ausführungen sind dieser Diplomarbeit entliehen. Die Fragebögen finden sich im Anhang.

5.2.2.3 Heuristische Evaluation mit Mock-Ups

Den Probanden wurde bei der Heuristischen Evaluation (HE) jeweils eine Serie von Screenshots gezeigt, die so präsentiert wurden, dass Arbeitsabläufe ersichtlich waren. Ein Screenshot aus der Mockup-Serie für die HE ist in Abbildung 5.1 zu sehen. Unter anderem wurde das im WOB-Modell vorgeschlagene Boolesche Eingaberaster zur Vereinfachung der Eingabe boolescher Logik in den Mock-Ups verwendet, um auch diese Eingabemöglichkeit zusammen mit dem Problem der *overconstraint Queries* auf Akzeptanz zu testen.

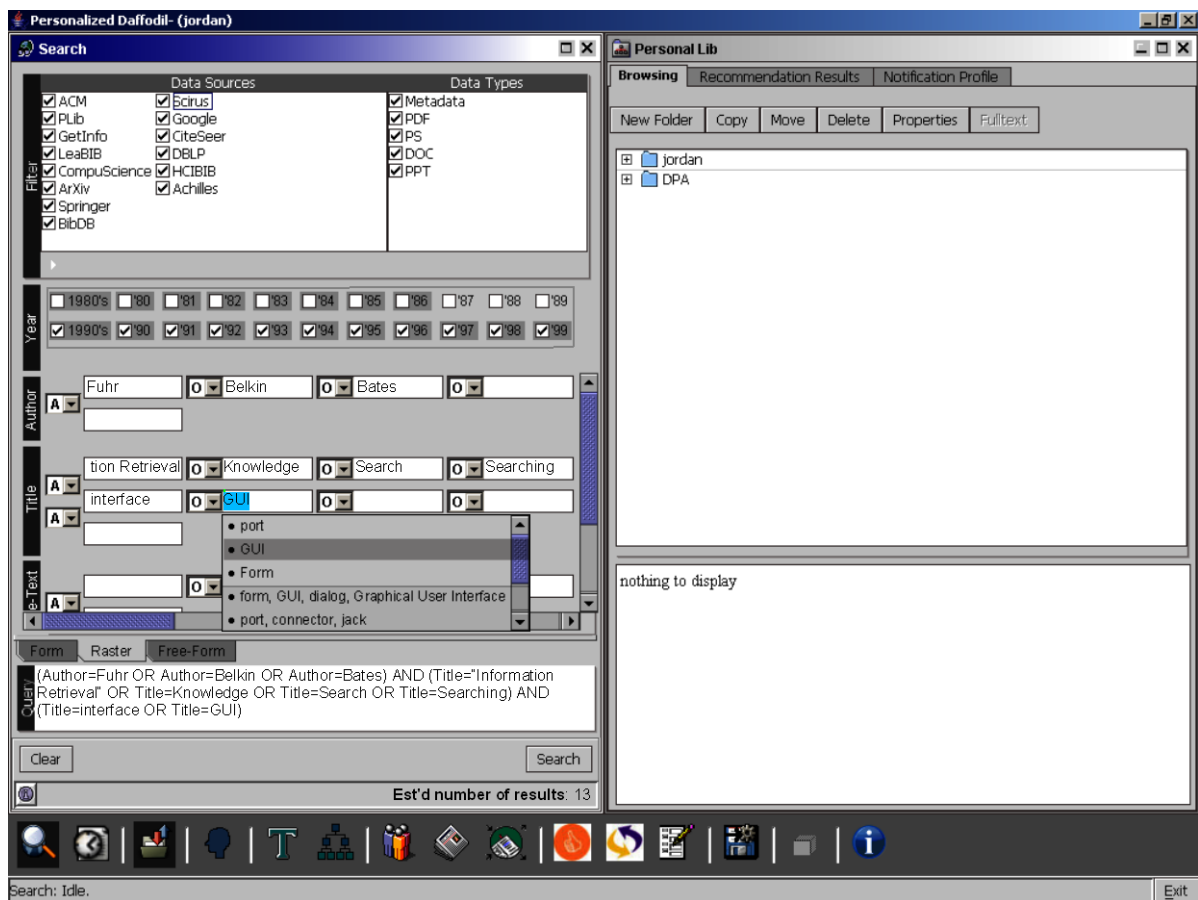


Abbildung 5.1: Screenshot Mockup für die heuristische Evaluation

Es sollte festgestellt werden, ob der Proband die jeweils dargestellten Vorlagen und Vorschläge versteht und diese nützlich findet.

Der Protokollführer erfragte, ob der jeweilige Vorschlag bei einer eigenen Recherche verwendet werden würde. Die Probanden gaben bei der Durchsicht der Entwürfe zu

Protokoll, ob sie Probleme sehen und ob Inkonsistenzen bemerkt wurden. Dabei sollten sie jeweils das Problem in eine heuristische Kategorie einordnen, wie von Nielsen (1994) beschrieben.

5.2.2.4 Ergebnisse der heuristischen Evaluation

Die Evaluatoren nannten, neben allgemeinen Usability-Problemen, folgende Probleme im Zusammenhang mit den proaktiven Vorlagen. Die Äußerungen sind den Mitschriften der Sitzung entnommen:

- Der Geltungsbereich der AND-Verknüpfung, beziehungsweise die Klammerung, ist nicht klar. (4 Nennungen)
- Verknüpfung zwischen den Abschnitten ist nicht klar.
- Die Operatoren-Rangfolge innerhalb einer Zeile ist unklar. Ggf. nur OR innerhalb einer Zeile und AND zwischen den Zeilen anbieten.
- Die Abkürzung von AND und OR durch „A“ und „O“ ist unklar. (3 Nennungen)
- Die Zustandsanzeige ist nicht auffällig genug.
- Die Zustandsanzeige ist uninteressant/ablenkend (bricht „minimales Design“).
- Die unvollständige Zeile am Ende eines Eingabebereiches (wie z.B. „Autor“) ist irritierend. (2 Nennungen)
- In Free-Form ist eine Hilfe sinnvoll.
- In Free-Form ist die Anzeige von Fehlern gewünscht.
- Konsistenz: Reihenfolge der Felder zwischen Form und Raster
- Konsistenz: Die ausgewählten Jahre werden in der Zustandsanzeige nicht dargestellt.

Darüber hinaus gaben die Teilnehmer noch folgende Hinweise zu anderen Bereichen des Benutzerinterfaces:

- Der Filter-Bereich hat weißen Hintergrund, als wäre es ein editierbares Textfeld. Der Hintergrund sollte grau sein.
- Die Benutzung des Filter-Bereichs ist unklar. Das betrifft die Auswirkung der Häkchen bei den Quellen als auch die der Dokumenttypen.

- Tooltips sollten für alle wichtigen UI-Elemente angeboten werden.
- Die Sprache des Interfaces ist Englisch, aber die meisten Benutzer sprechen Deutsch als Erstsprache.
- Der Filterbereich könnte eingefaltet und alle Checkboxes per Default gesetzt werden.

Die Heuristische Evaluation zeigte, dass die Vorlagefunktionen akzeptabel waren, einige Aspekte mussten vor der finalen Implementation überarbeitet werden. Meist bezogen sich die Änderungen auf Form und Farbgebung der Fehlermarkierungen, sowie auf Hinweise, die zu weit weg vom aktuellen Fokus der Aufmerksamkeit dargestellt wurden. Die Markierungen direkt im Formulareingabefeld wurden allgemein akzeptiert.

5.2.2.5 Kontrollierte Benutzertests in zwei Gruppen

Basierend auf dem überarbeiteten Design nach der Heuristischen Evaluation wurde der erste funktionale Prototyp implementiert und nachfolgend getestet. An dem Test nahmen 20 Informatikstudenten teil. Die Studenten waren noch im Vordiplom und hatten entsprechend wenig Erfahrung mit wissenschaftlicher Recherche.

Diesen Studenten wurden fünf Aufgaben gestellt, die sich zwecks Vergleichbarkeit an den Aufgaben aus Abschnitt 5.2.1.2 anlehnten. Die Aufgabe war somit eine Mischung aus *Known Item Instantiations* und themenbasierten Suchen. Die Aufgaben waren analog zu den Aufgaben bei der Evaluation zur strategischen Unterstützung (s.o.):

Aufgaben zur Evaluation von Daffodil

Aufgabe 1

Du hast gehört, dass es einen guten englischen Artikel gibt über Normal-Formen in der relationalen Datenbank-Theorie. Der Artikel soll aber schon etwas älter sein. Finde diesen Artikel.

Aufgabe 2

Suche nach Artikeln von Eric Horvitz der letzten 4 Jahre aus dem Themenbereich "User Interfaces".

Aufgabe 3

Mit welchem Hauptthema beschäftigt sich Prof. Joachim Biskup?

Aufgabe 4

Finde alle Artikel von Eric Horvitz zusammen mit Susan Dumais der letzten 4 Jahre.

Aufgabe 5

Alternative a)

Du möchtest Dich in die Evaluation von Webseiten und digitalen Bibliotheken aus Sicht des User-Interface-Designs einarbeiten. Finde eine moderne Methode in diesem Bereich.

Alternative b)

Finde Informationen über Cybersickness. Du interessierst Dich speziell für die Häufigkeit und die Situationen des Auftretens.

Die Probanden wurden nach der Recherche gebeten, einen Fragebogen auszufüllen. Der Fragebogen ist in Anhang A zu finden, in der Variante für die Baseline-Evaluation, ohne proaktive Funktionen, sowie für die Gegenprobe mit proaktiven Funktionen.

Ergebnisse Die Anfragen, die an das System zur Evaluation gesendet wurden, wurden in Log-Dateien aufgezeichnet und analysiert. Da es fünf kontrollierte Aufgaben zu bearbeiten gab, gab es einen hohen Grad an ähnlichen Anfragen. Betrachtet wurden die 246 Queries, die während der Evaluations-Sitzungen mit den Studierenden gesendet wurden. 142 der Queries stammen aus dem Test mit der Baseline und 104 aus dem Test mit der um proaktive Funktionen erweiterte Version von Daffodil.

Die Anzahl der allgemein fehlerhaften Queries (mit Ausnahme der lediglich mehrfach gesendeten Queries, die sonst keinen weiteren Fehler enthalten und damit nicht per se falsch sind), liegt insgesamt bei 94 (42 bzw. 52). Damit sind insgesamt 38% der Queries fehlerhaft.

Es gab einige besonders häufige Fehlerklassen:

- Orthographische Fehler in Autorennamen und Suchtermen
- Ungültige Anwendung von logischen Operatoren oder andere syntaktische Fehler

5 Evaluation

- Unsachgemäße Anwendung der Suchsprache (z.B. die Verwendung deutscher Suchbegriffe um englische Dokumente zu finden)
- Überspezifizierte Anfragen – Anfragen ohne Antwort, weil nicht alle Teile der mit dem logischen AND verknüpften Anfrageteile gemeinsam auftraten
- Unterspezifizierte Anfragen – Anfragen mit sehr hochfrequenten Termen und einer dadurch viel zu großen Treffermenge oder Anfragen nur mit Angaben zum Zeitraum der Veröffentlichung, aber ohne weitere Suchterme

Die durchschnittliche Anzahl der Anfragen der Gruppe ohne aktive Vorlage war vergleichbar mit der Gruppe, bei der diese Funktionen aktiviert waren. Die jeweiligen Zahlen sind in der Tabelle 5.2 wiedergegeben.

	Task 1	Task 2	Task 3	Task 4	Task 5a	Task 5b	Average
User 1	3	1	3	2		1	2,00
User 3	6	3	4	4		7	4,80
User 4	6	1	2	1		5	3,00
User 5	7	2	1	6		3	3,80
User 6	9	1	5	4		7	5,20
User 7	3	3	1	2	5		2,80
User 8	3	2	1	19		8	6,60
Average	5,29	1,86	2,43	5,43	5,00	5,17	4,03
User 12	4	3	10	3			5,00
User 14	6	4	4	3	3		4,00
User 15	6	3	3	1		1	2,80
User 16		10	3	1			4,67
User 17	6	2	10	2			5,00
Average	5,50	4,40	6,00	2,00	3,00	1,00	4,29
Average	5,36	2,84	3,80	4,11	4,33	4,65	4,10

Abbildung 5.2: Anzahl der Anfragen je Nutzer und Aufgabe

Die zweite Gruppe hat die Recherche mit wesentlich weniger Rückfragen und weniger negativen Kommentaren durchgeführt. Aus der Beobachtung lässt sich schließen, dass die Defizite bei der Strategie der Recherche, also die Informationskompetenz, für den Gesamterfolg entscheidender war, als die Einzelfehler bei den Anfragen, die meist bei den iterativen Anfrageverfeinerungen ausgeglichen wurden. Diese Beobachtung ist konsistent mit den Ergebnissen bei Brajnik u. a. (2002), die berichten, dass Suchende oft bei syntaktischen oder terminologischen Problemen Hilfe erfragten, sich jedoch ihrer *strategischen Defizite* nicht bewusst waren.

5.2.2.6 Benutzertests mit Videoprotokoll

Nach der Auswertung des Benutzertests, wie im letzten Abschnitt beschrieben, wurde die Implementation noch einmal verfeinert.

Mit dieser dritten Implementation wurden Einzeltests mit Videoaufzeichnung durchgeführt. Die Probanden waren dabei aufgefordert, laut über ihr Vorgehen und ihre Beobachtungen zu berichten. Die Aufgaben waren auch bei diesem Test die selben wie in dem Gruppentest zuvor. An dem Test nahmen insgesamt zehn Personen teil. Neun davon waren wissenschaftliche Mitarbeiter der Universität Duisburg-Essen und hatten eine relativ hohe Informationskompetenz. Zudem nahm an diesem Test eine Bibliothekarin teil, die hauptberuflich auch Weiterbildungen zu Recherchestrategien bei OPAC-Recherchen anbot. Von dieser Gruppe wurde ein differenziertes Meinungsbild zu den angebotenen Vorschlägen und Hilfestellungen erwartet.

Bei den Tests wurde insbesondere beobachtet, wie das zeitliche Verhalten der Funktionen wirkt. Die Vorlagen erschienen nach einer kurzen Verzögerung, da davon ausgegangen wird, dass Nutzer, die noch bei der Eingabe sind und keine Tipp-pausen machen, keine Vorlagen erwarten und abgelenkt würden. Die individuellen Erwartungen bezüglich der zeitlichen Parameter waren jedoch sehr unterschiedlich. Einige Nutzer hätten eine Gestaltung bevorzugt, welche die Aufmerksamkeit stärker auf sich zieht und länger am Bildschirm sichtbar bleibt. Das Videoprotokoll belegt, dass die meisten Nutzer ohnehin bei der Eingabe auf die Tastatur schauen und nicht auf den Bildschirm. Ein Benutzer lehnte Vorlagefunktionen prinzipiell ab und wollte die Vorlagen lieber abstellen – was durch die Personalisierungsoptionen möglich war. Zwischen den diversen Anforderungen schien die gewählte Verzögerung von circa einer Sekunde bis zum Erscheinen einer Markierung oder Vorlage ein akzeptabler Kompromiss zu sein.

Die generelle Akzeptanz der Vorlagefunktionen variierte: Während einige Benutzer die Vorlagen akzeptierten und generalisierten (sie regten weitere Vorlagen und aktive Hilfe in einigen Situationen an), verweigerten andere die Verwendung der Vorlagen. Der Schluss hieraus ist, dass es möglich bleiben muss, die proaktiven Funktionen zurückzunehmen, wenn dies den persönlichen Vorlieben entspricht.

Einige Detail-Ergebnisse sind im Folgenden thematisch sortiert wiedergegeben.

Popup-Liste Die Popup-Liste mit den Vorlagen scheint zurückhaltend genug zu sein, dass Benutzer die Vorschläge ungestört übergehen können. Tatsächlich antwortete sogar eine Testperson auf die Frage, ob er Hilfsangebote von DAFFODIL gesehen habe, mit „nein“, obwohl die Liste kurz aufgeklappt war. Das ist ein gutes Indiz dafür, dass die Popupliste zumindest keine Frustration auszulösen scheint. Andere Testpersonen äußerten auch explizit, dass die Liste nicht störe.

Problematisch war allerdings, dass die Popupliste zu zurückhaltend war: Einige Benutzer bemerkten die Vorlagen erst recht spät während der Evaluations-Sitzung, weil sie vorher immer Wörter vollständig ausgeschrieben und ein Leerzeichen angefügt haben, um weitere Terme hinzuzufügen. Wie oben bereits beschrieben, legt DAFFODIL nur für das gerade am Cursor befindliche Wort Vorschläge vor, um den Benutzer bei der Eingabe in der *aktuellen Situation* zu unterstützen. Während diese Policy beim Editieren mitten in der Eingabezeile sinnvoll ist, ist sie am Ende der Zeile offensichtlich zu restriktiv.

Ein anderer Grund dafür, dass einige Testpersonen die Liste nicht bemerkt haben, war, dass sie bei der Eingabe der Suchanfrage auf die Tastatur geschaut haben. In diesem Fall sehen die Benutzer die Popup-Liste oft selbst dann nicht, wenn die Liste länger aufgeklappt bleibt.

Aufruf, Navigation der Popup-Liste Bei vier von sieben Testpersonen konnte man beobachten, wie sie die Vorlage absichtlich provozierten, indem sie ein Wort änderten und den Timeout abwarteten. Allerdings konnte nur bei einer Testperson beobachtet werden, dass sie die Vorlagen mittels Tastenkombination aufruft. Diese Beobachtung ist in sofern erstaunlich, als dass es einen entsprechenden Hinweis in den Tooltips gibt und Benutzer wiederholt die Tooltips durchgelesen haben. Eine denkbare Erklärung dafür ist, dass die Benutzer die Tooltips nur gelesen haben, bis ihre Frage beantwortet war, und dass ihre Fragen sich meistens auf die Syntax in den Formularfeldern bezog. Der Hinweis auf die Tastenkombination befindet sich aber am Ende des Tooltips.

Alle Benutzer haben Einträge in der Vorlageliste mit der Maus ausgewählt, die Liste aber mit Druck auf die Escape-Taste geschlossen, falls sie keinen Eintrag auswählen wollten. Das ist bemerkenswert, da auf diese Möglichkeit nicht hingewiesen wurde.

Obwohl die Liste auch mit den Cursortasten navigierbar ist, hat keine der Testpersonen diesen Versuch unternommen.

Markierung von Termen Probleme gab es mit der Unterschlängelung von Termen in den Textfeldern. Die proaktiven Funktionen markieren Terme, zu denen sie eine Vorlage machen können, mit einer roten Unterschlängelung, wenn der betreffende Term vermutlich fehlerhaft ist, oder in blau, wenn der Term nicht fehlerhaft ist, aber z.B. durch Synonyme ergänzt werden kann. Keine der Testpersonen konnte die Bedeutung der blauen Markierung auf Anhieb nennen. Einige Testpersonen vermuteten, dass die blaue geschlängelte Linie „wie bei Word“ auf ein falsch geschriebenes Wort hinweist. Nur eine Testperson konnte nach einiger Zeit und nachdem sie auch die rote Markierung gesehen hatte, den Unterschied in der Bedeutung beider Markierungen erklären.

Markierung von Feldern Auch die roten Klammern um Felder mit Syntax-Fehlern herum waren offensichtlich missverständlich. Ein Benutzer mutmaßte, die Klammern zeigen das Feld an, in dem gerade editiert wird. Diese Beispiele zeigen, dass Farbsymboliken mit äußerster Vorsicht zu verwenden sind und, wo möglich, durch Text oder Icons ersetzt oder ergänzt werden sollten.

Transparenz der Vorschläge Was die Präsentation der Vorschläge angeht, hat ein Benutzer die angebotenen Terme der Related-Terms-Vorlage mit dem Kommentar „mir ist nicht klar, woher diese Terme kommen und wieso die mir angeboten werden“ geschlossen. Hier scheint es sinnvoll zu sein, die Vorlage sinnvoll zu kommentieren - und zwar über die reine Feststellung, es handele sich um einen „Related Term“ hinaus.

Sprachprobleme Ein Problem, das in allen Evaluationsphasen auftrat, ist, dass keine der Testpersonen Englisch als erste Sprache hatte, die überwiegende Mehrzahl der Texte, die DAFFODIL erreichen kann, aber in Englisch verfasst sind und die Benutzeroberfläche des Systems Englisch ist. Darüberhinaus unterstützt DAFFODIL keine multilinguale Suche; d.h. Suchbegriffe müssen in der Sprache verwendet werden, in der auch die zu findenden Dokumente verfasst sind. Häufige Fragen in diesem Zusammenhang waren „was heißt ‚Normal-Form‘ auf Englisch?“ oder „was heißt ‚Häufigkeit‘ auf Englisch?“.

Namen in unterschiedlichen Kulturräumen Eine spannende Beobachtung war die Schwierigkeit, die eine Testperson mit Arabisch als erster Sprache bei der Spezifizierung des Autoren-Namens hatte: Der Testperson war unklar, welcher Teil des Namens, nach der sie suchen sollte, der Familienname war. Im Arabischen, wie auch in einigen asiatischen Sprachen, wird erst der Familienname und dann der persönliche Name eines Menschen genannt. In westlichen Sprachen ist die Reihenfolge in der Regel umgekehrt, also in der Reihenfolge *Vorname Familienname*. Da in der Übersetzung mancher Texte auch die Reihenfolge der Namensbestandteile an die der Zielsprache angepasst wird, aber in anderen wiederum nicht, besteht hier eine Mehrdeutigkeit.

Reformulierung Eine Reformulierung, die mehrfach vorkam, war die Übernahme der Terme aus dem Title-Feld in das Free-Text-Feld und umgekehrt. Eine Bewegung der Suchterme vom Title-Feld zum Free-Text-Feld entspricht einer Erweiterung der Suche, da Terme im Free-Text-Feld sowohl im Titel des Dokuments als auch im Abstract und evtl. sogar im Volltext gesucht werden. Die entgegengesetzte Bewegung entspricht daher einer Einengung der Suche. Diese Taktiken könnte ein proaktives Modul vorschlagen, wenn ein Schwellenwert an Suchtreffern über- oder unterschritten wurde, und auf Benutzerwunsch durchführen.

Nutzwert der Vorlagen Alle befragten Personen äußerten, dass ihnen die Vorlagen geholfen haben. Testpersonen sagten, die Vorlagen „erleichtern die Tipp-Arbeit“, „sind vermutlich hilfreich für Suchende, die sich im Thema nicht so gut auskennen“ und „der Spell-Checker spart Zeit und gibt Sicherheit“.

Die meisten Anwender nutzten die Vorlagen einige Male in den frühen Phasen der Recherche. Die Tendenz zur Nutzung nahm ab, je zuversichtlicher die Anwender waren, dem gewünschten Ergebnis näher zu kommen.

Die Funktionen greifen proaktiv direkt in den benutzergesteuerten Prozess ein. Während die meisten Teilnehmer die angebotene Hilfe annahmen – insbesondere in der frühen Recherchephase, die am stärksten durch Unsicherheit belastet ist – reagierten einige Personen darauf mit Ablehnung. Dies bestätigt, dass proaktive Vorlagefunktionen personalisierbar und abschaltbar gehalten werden sollten, da nicht alle Anwender die gleiche Akzeptanz mitbringen.

5.2.2.7 Zufriedenheit der Testpersonen

5 Evaluation

Das Ergebnis der allgemeinen Frage nach der Zufriedenheit der Testperson mit DAFFODIL ist in Abbildung 5.3 zusammengefasst. In dieser Abbildung markiert ein Punkt eine Antwort einer Testperson und das Dreieck markiert den Mittelwert aller Nennungen.

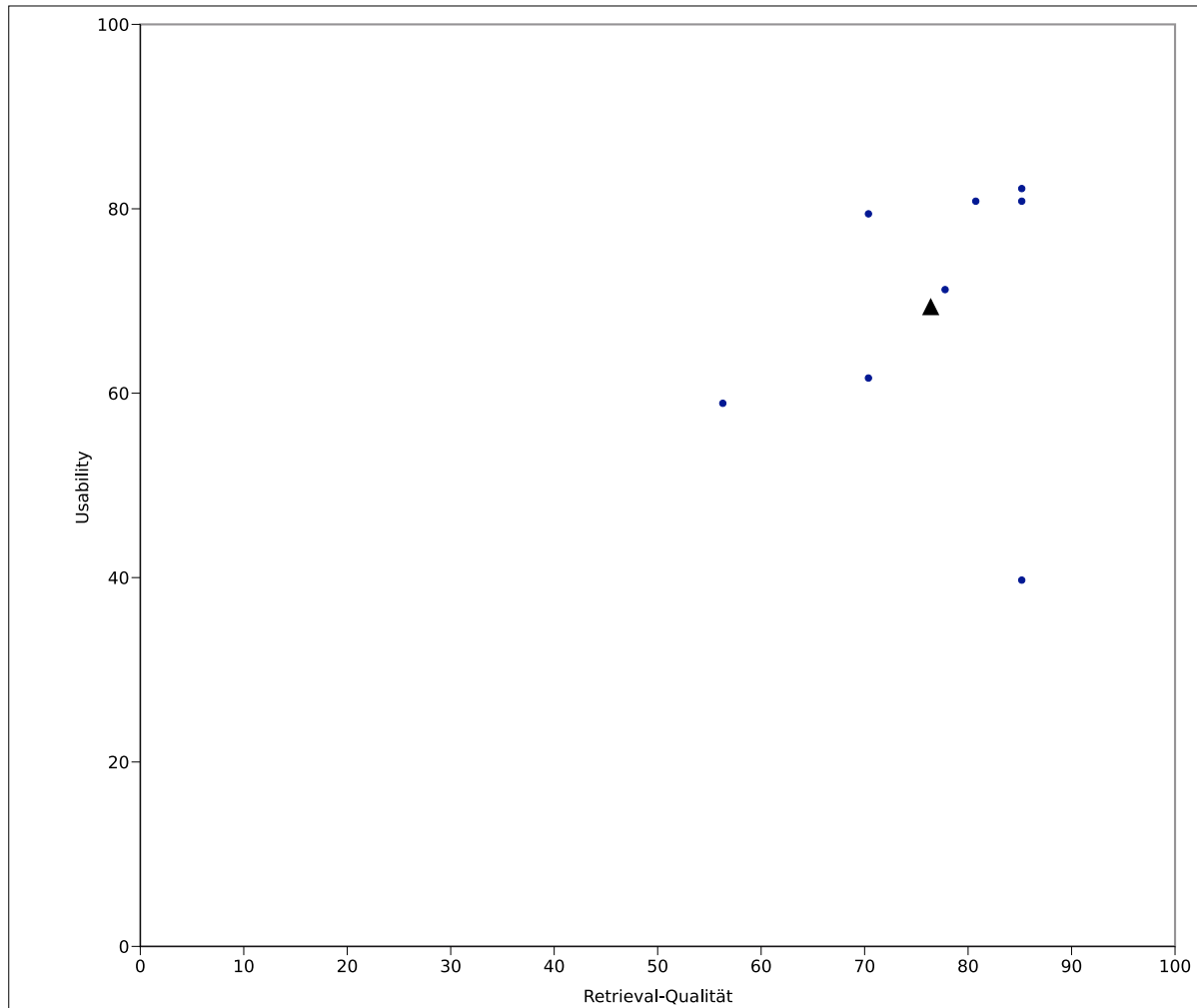


Abbildung 5.3: Zufriedenheits-Ebene

Mit Ausnahme einer Testperson haben alle Testpersonen in der jeweils oberen Hälfte der Skala bewertet. Da die Frage auf eine allgemeine Bewertung von DAFFODIL ausgerichtet war, ist dies ein Hinweis darauf, dass die Entwicklung von DAFFODIL insgesamt auf dem richtigen Weg ist.

6 Fazit

Zu Beginn dieser Arbeit stand das Problem der verteilten Verfügbarkeit immer größerer Informationsbestände im Fokus der Analyse. In Hinblick auf die wissenschaftliche Recherche in einem Fachgebiet bietet sich dem Nutzer schon von seinem Arbeitsplatz aus eine Fülle von Informationsangeboten, deren Inhalte aber bisher weitgehend isoliert nebeneinander existieren. Tiefe Verknüpfungen auf verschiedenen Handlungsebenen sind notwendig für die tatsächliche Nutzung des Potentials an Mehrwert, die durch die Fülle an Informationen in Reichweite scheint. Die Vision Vannevar Bushs, und auch dessen Problematisierung, in „How we may think“ einer Maschine, die uns hilft, den Überblick über die Informationsflut zu erhalten, begleitet die Idee des agentengestützten Informationssystems.

Die Unterstützung des Anwenders in Hinblick auf Informationskompetenz und Recherchestrategie ist eine Notwendigkeit für ein zufriedenstellendes Ergebnis; das zeigen sowohl die theoretische Analyse, als auch die empirischen Studien, die die Konzeption und Implementation des ganzheitlichen Informationssystems DAFFODIL begleitet haben.

Enge Integration der Informationsquellen und strategische Unterstützung durch höhere Suchfunktionen erleichtert das Erreichen der Ziele des Nutzers. Die Unsicherheit bei der Recherche, die sich aus dem *Anomalous State of Knowledge* ergibt, kann durch Vorschläge und Hilfestellungen, die nah am Kontext des Anwenders sind, reduziert werden. Dennoch reicht für die Zufriedenheit des Anwenders nicht die alleinige Unterstützung der Recherche. Er muss auch beim Management der Komplexität unterstützt werden und bei allen Arbeitsabläufen die Orientierung behalten können. Dazu dienen die diversen Werkzeuge, die neben der föderierten Suche angeboten werden. In Werkzeugen – wie der Koautorensuche und der kollaborativ nutzbaren und sich dynamisch anreichernden persönlichen Handbibliothek – manifestieren sich Mehrwertdienste, die zur Arbeitsumgebung bei der erfolgreichen Recherche beitragen.

Aus Sicht der Software-Ergonomie wird der Frage nachgegangen, wie der Zielkon-

flikt zwischen proaktivem Verhalten der Software-Agenten zur Unterstützung der Anwender und deren grundlegendem Bedürfnis nach Handlungsautonomie aufgelöst werden kann. Ausgehend von Betrachtungen zur Metaphorik und diversen bekannten Verwendungsmustern von Software-Agenten wird folgender Schluss gezogen: Nicht die aufwändige Vermenschlichung von Software-Agenten ist der Schlüssel für gelungene Hilfestellung, sondern die weitgehende Bewahrung der Benutzerautonomie und die zurückhaltende aber stets verfügbare Markierung von Möglichkeiten und die geschickte, unaufdringliche Präsentation von Vorlagen.

Software-Agenten sind intelligente Komponenten, die im WOB-Modell verankert sind und teilen sich aus dem Hintergrund die Arbeitsumgebung mit dem Anwender als Wirkungsbereich. Auf eine Modellierung der Benutzerintention und tiefgreifende künstliche Intelligenz muss dabei nicht in erster Linie gesetzt werden. Wichtiger erscheint die Beobachtung einschlägiger Handlungen und die deduktive Herleitung passender Informationen. Hierbei hilft das Monitoring, also laufende, automatisierte Recherchen in Bezug auf ein vom Benutzer definiertes Thema, und Awareness, die Präsentation und Hervorhebung neuer Information, die dynamisch der persönlichen Handbibliothek zugefügt wurde.

Auch bei der föderierten Integration verteilter Datenquellen können Software-Agenten die Aufgaben der Integration, der Homogenisierung von Metadatenschemata und der damit verbundenen semantischen Heterogenitätsbehandlung und das Filtern und Relevanzbewerten zusammengeführter Resultate übernehmen. Die prototypische Implementation im Rahmen von DAFFODIL belegt, dass dies möglich und produktivitätssteigernd ist.

Je weiter die mögliche Automatisierung fortschreitet, desto genauer muss auf die Handlungsautonomie des Anwenders geachtet werden, der Ziele verfolgt, die dem Softwaresystem verborgen bleiben. Auch die begrenzte Resource der Aufmerksamkeit ist der Software nicht zugänglich und somit wird die Vermutung bestätigt, dass übertriebene Automatisierung kontraproduktiv sein kann.

Die in der DAFFODIL-Software vorgenommene vertikale Implementierung der erfolgversprechenden Konzepte in Bezug auf die Arbeitsteilung zwischen Informationssystem und Anwender – auf allen Ebenen – und die durchgängige Begleitung durch empirische Evaluation dient als Proof of Concept und ruft zur ganzheitlichen Herangehensweise an fachbezogene Literaturrecherchesysteme vor dem Hintergrund der aktuellen Informationsfülle auf.

Die Ergebnisse zeigen: Das System kann nicht nur von Experten genutzt werden, sondern unterstützt auch unerfahrene Anwender bei der erfolgreichen Recherche. Bei steigender Komplexität der Informationslandschaft ist die Steigerung der Informationskompetenz für Nicht-Experten dringend notwendig.

Die Ergebnisse der vorliegenden Arbeit sind somit einerseits die detaillierte Analyse des Konfliktpotentials, welches durch das Aufeinandertreffen zweier Kontrollparadigmen entsteht, nämlich der direkten Manipulation versus der Delegation und Automatisierung mittels Softwareagenten. Das zweite Hauptergebnis ist die Erarbeitung einer plausiblen Lösung dieses Konfliktes: Der Konflikt, der im Bereich der Informations- und Recherchesysteme besonders heikel ist, lässt sich kaum vermeiden, da der fortschreitende Einsatz von Agenten und intelligenten Softwarekomponenten zur Bezwingung der steigenden Informationsflut notwendig geworden ist. Die Herausarbeitung der Notwendigkeit, das Autonomiebedürfnis der Anwender zu respektieren und die Agentendienste als höherwertige Dienste und Werkzeuge anzubieten, wie es das WOB-Modell ermöglicht, erlaubt eine weitgehende dialektische Auflösung des scheinbaren Widerspruchs. Der Anwender behält seine Kontrolle und maximale Handlungsfreiheit, kann jedoch auf weitreichende Integration von Informationsquellen und automatisierte Unterstützung durch Agenten und Dienste zurückgreifen.

A Evaluation

A.1 Der Fragebogen für die Evaluation der Baseline

A.1.1 Vor jeder Aufgabe

- Kannst Du mit der Aufgabe etwas anfangen? Wie sicher fühlst Du Dich bei dieser Suche? Bitte ankreuzen:
Sehr sicher (1) (2) (3) (4) (5) sehr unsicher

A.1.2 Nach jeder Aufgabe

- Gab es Punkte, an denen du bei der Formulierung der Suchanfrage unsicher warst? Wenn ja: welche Punkte waren das?
- An den Stellen, an denen du unsicher warst: welche Fragen hättest Du einen Experten gefragt, wenn du gekonnt hättest?
- Gemessen an deiner Erwartung: wie gut konntest Du die Aufgabe lösen?
Wesentlich besser (1) (2) (3) (4) (5) wesentlich schlechter

Weiterhin wurden Start- und Endzeit festgehalten.

A.2 Der Fragebogen für die Evaluation der proaktiven Version von Daffodil

A.2.1 Vor jeder Aufgabe

- Kannst Du mit der Aufgabe etwas anfangen? Wie sicher fühlst Du Dich bei dieser Suche? Bitte ankreuzen:
Sehr sicher (1) (2) (3) (4) (5) sehr unsicher

A.2.2 Nach jeder Aufgabe

- Wie sicher warst Du bei der Suche? Bitte ankreuzen:
Sehr sicher (1) (2) (3) (4) (5) sehr unsicher
- Wie zufrieden bist du mit den Suchergebnissen? Bitte ankreuzen:
Sehr zufrieden (1) (2) (3) (4) (5) sehr unzufrieden
- Welche Vorlageleistungen hast Du gesehen?
- Was vermutest Du, welche Hilfe Dir die Vorlageleistungen bieten sollten?
- Hast Du die Vorlageleistung benutzt? Welche?
- Haben die Vorlagen Deine Arbeit beeinflusst? Wie?

Weiterhin wurden Start- und Endzeit jeder Aufgabe festgehalten.

A.2.3 Am Ende der Sitzung

Folgende Fragen wurden der Testperson nach Beendigung der letzten Aufgabe gestellt:

- Hast Du die Rechtschreibprüfung gesehen? Wenn ja: welchen Einfluss hatte sie auf dich (hilfreich, störend, verwirrend, irreführend)? Begründe bitte.
- Hast Du die Queryhistory benutzt? Wenn ja: welchen Einfluss hatte die History auf dich? Welchen Einfluss hatten die Zusatzinformationen in der Liste auf Dich (hilfreich, störend, verwirrend, irreführend)? Begründe bitte
- Hast Du die Autorenergänzung gesehen? Wenn ja: welchen Einfluss hatten die Vorschläge auf Dich?
- Hast Du die Termvorschläge gesehen? Wenn ja: welchen Einfluss hatten die Vorschläge auf Dich?
- Hast Du die Augenmarkierungen bei den Einträgen der Ergebnisliste gesehen? Wenn ja: was glaubst Du, soll das bedeuten? Hilft Dir diese Information?

A Evaluation

Außerdem wurde jede Testperson gebeten, in einer Zufriedenheitsebene einen Punkt zu markieren. Der Wert auf der X-Achse stand dabei für die Retrieval-Qualität und der Wert auf der Y-Achse für die Usability.

B Kurzübersicht Klassenbibliothek

B.1 WOB-Werkzeuge und Desktop

Aufbauend auf den allgemeinen Oberflächenkonzepten des WOB-Modells (siehe Abschnitt 4.3) ist eine Klassenbibliothek erstellt worden, die grundlegende Klassen zur Erstellung von zum WOB-Modell konformen Oberflächen enthält.

Ein weiterer wichtiger Baustein ist die nachrichtenbasierte Kommunikation zwischen den Werkzeugen.

Das WOB-Modell definiert Gestaltungsrichtlinien zum Entwurf von Informationssystemen. Das zentrale Konzept sind *Werkzeuge*, die Informationsobjekte ver-/bearbeiten. Die Werkzeuge sollen so miteinander kombinierbar sein, dass auf eine komplexe Menüführung weitgehend verzichtet werden kann. Alles soll durch Drag'n'Drop, Nachrichtenaustausch zwischen den Werkzeugen oder das Verfolgen von Links miteinander verknüpft werden können.

B.1.1 Überblick

Die Elemente der DAFFODIL-Oberfläche fallen in die folgenden Klassen:

1. *Desktop*
Hauptbildschirm
2. *Views*
Die geöffneten Ansichten der Werkzeuge, zum Beispiel Such- oder Ergebnisbildschirm oder die Ansicht der persönlichen Handbibliothek.
3. *Tools*
Die Werkzeuge, zum Beispiel das Suchobjekt, der Thesaurus-Browser oder die Handbibliothek.

B.1.2 Anforderungen

Bei den Tools handelt es sich um Werkzeug-Objekte, die in einem Container (Toolbar bzw. Werkzeugleiste) immer verfügbar sind. Die Menge ist offen. Tools können dynamisch hinzugefügt oder entfernt werden.

Ein Tool kann *ein oder mehrere Ansichten (Views)* besitzen. Der View öffnet sich bei Klick auf das Tool, oder wenn das Tool durch andere Events aktiviert wird. Die Objekte auf der Werkzeugleiste können nicht aus der Toolbar herausgezogen werden. Objekte (Dokumente, Autorennamen, Klassifikationseinträge) können auf die Tools gezogen werden. Wenn ein Tool eine Funktion für das Objekt anbietet, signalisiert es das durch eine grüne Umrandung.

Durch Drag and Drop aus den Ansichten auf die Ansicht des speziellen Werkzeugs *Handbibliothek* können die Objekte in einem persönlichen Bereich abgelegt und verwaltet werden.

Weitere Anforderungen

- Loses koppeln der Oberflächenelemente, basierend auf Event und Nachrichtenaustausch. Elemente sollen möglichst einfach gegen ähnliche Komponenten getauscht werden können.
- Globale Funktionen (z.B.: Schließen aller Fenster) sollen über Nachrichtenaustausch möglich sein, sodass sie von neu hinzukommenden Werkzeugen auf einfache Weise umgesetzt werden können.
- Kontextmenüs sollen überall da zur Verfügung stehen, wo es zur Arbeitserleichterung um abkürzende Aktionen geht, oder wenn Nachrichten zu dem gewählten Kontext ausgelöst werden sollen.

B.1.3 WOB-Bibliothek

Die Kernklassen der WOB-Bibliothek sind `Tool`, `View` und `Desktop`.

Es ist mit dieser Bibliothek möglich, Werkzeuge und Ansichten durch Ableitung von einer abstrakten Klasse zu definieren und diese Werkzeuge auf einer `Toolbar` anzeigen zu lassen. Die Ansichten der Werkzeuge öffnen sich als `InternalViews` innerhalb des Desktop Fensters, im Sinne der MDI-Philosophie. (Siehe B.1)

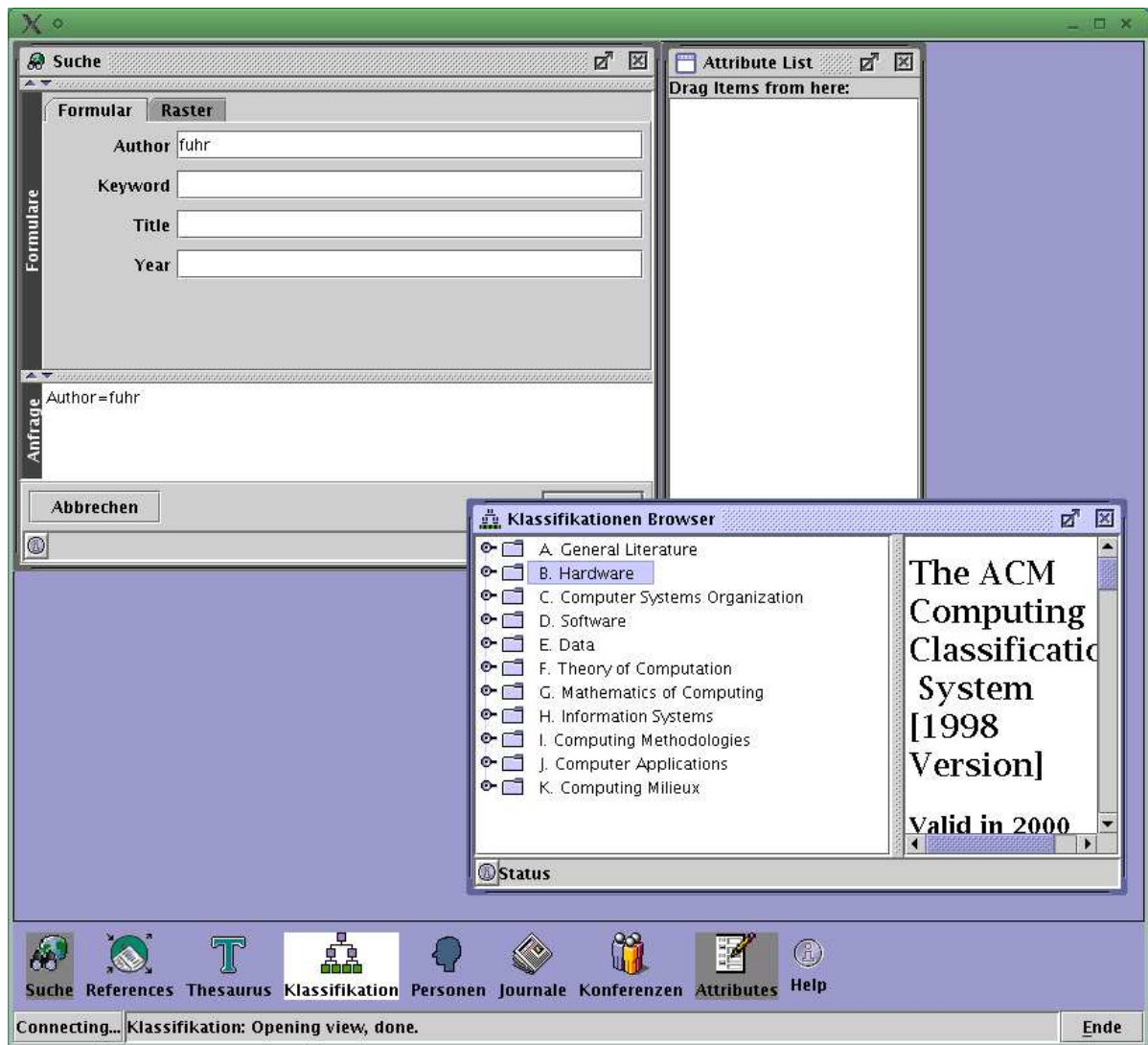


Abbildung B.1: WOB Desktop, Version 1

Die Klasse `DefaultApp` ist ausführbar. Diese ist jedoch nur zu Testzwecken vorhanden. Auf der Bibliothek aufbauende Applikationen implementieren eigene `*App` Klassen (zum Beispiel: `DaffodilApp` oder `PaddleApp`).

B.2 Klassenmodell

In Abbildung B.2 sind die wichtigsten Klassen der Bibliothek und ihre Relationen zu erkennen: `Tool` und `ToolView` definieren die grundlegenden Eigenschaften der Werkzeuge und ihrer Ansichten. Überall wo in späteren Anwendungen auf ein `Tool` oder seinen `View` Bezug genommen werden soll, sollten diese beiden Interfaces verwendet werden.

Wichtige Eigenschaften von Schnittstellen implementierender Klassen sollten durch Erweiterung von `Tool` und `ToolView`, bzw. deren Spezialisierung (Ableitung) reflektiert werden.

`DefaultTool` und `DefaultView` sind konkrete Ableitungen zu Demonstrationszwecken.

`ToolController` ist eine Kontrollinstanz (Singleton) für alle `Tool`-Instanzen – über diesen Controller werden die Werkzeuge in die Infrastruktur der Arbeitsoberfläche und deren Nachrichtenkommunikation eingebunden.

B.2.1 Besonderheiten

MDI-Tools: Zwischen der Interface-Abstraktionsebene (`Tool` & `ToolView`) und der Implementationsebene gibt es noch die abstrakten Klassen `InternalTool` und `InternalView`. In diesen wird gemeinsames Verhalten von MDI-Werkzeugen und Ansichten definiert. Statt die `Tool`- und `View`-Funktionalität in gemeinsamen abstrakten Klassen auszudrücken, müssen voraussichtlich je ein Paar für externe wie für interne Werkzeuge *nebeneinander* stehen.

Die `Tools` erben bereits das `EventListener`-Interface. Damit sind alle `Tools` potentielle Empfänger von Event-Nachrichten über die dafür entwickelte Infrastruktur (siehe Abschnitt B.4). Demonstriert wird dies in der Beispiel-Applikation über ein Kontextmenü, das zum Desktop-Hintergrund gehört. Wird der Eintrag „Send Message“ aktiviert, werden an alle `Tools` Nachrichten vom Typ `WobEvent` über den `Dispatcher`

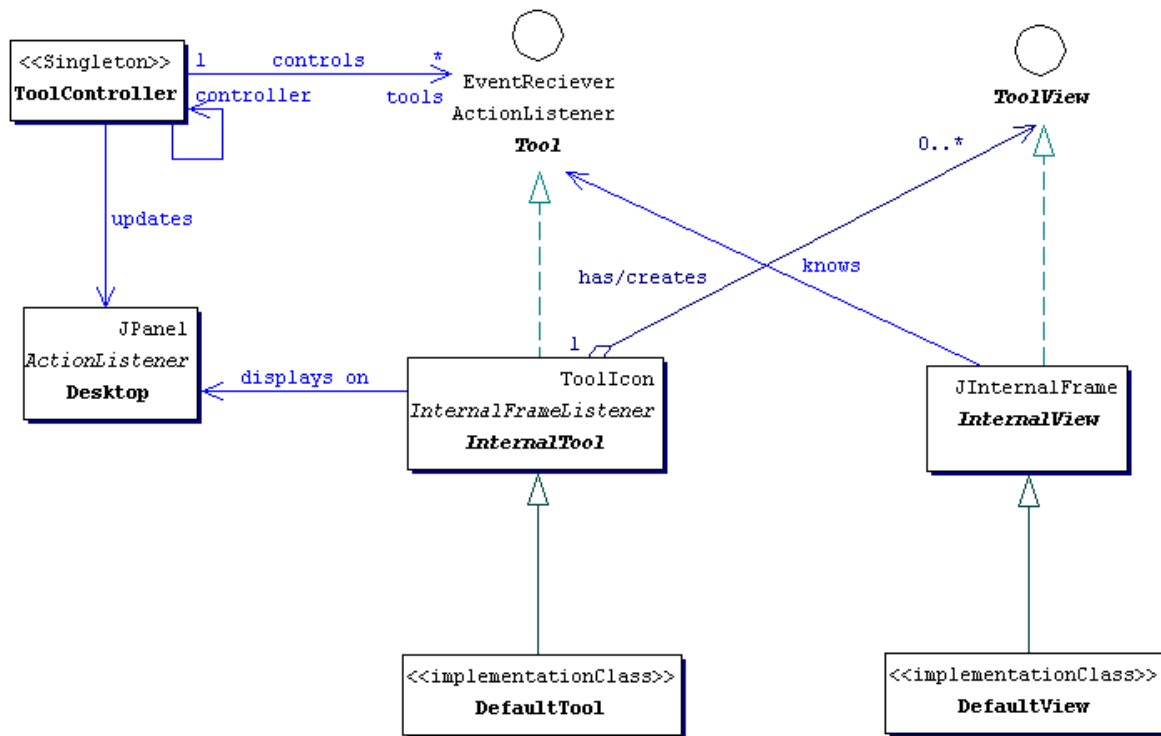


Abbildung B.2: Klassen im Desktop

verschickt. Jeder Empfänger antwortet im Beispiel mit einer Ausgabe auf der Kommandozeile.

B.3 Anwendungskontexte

B.3.1 Neue Tools definieren

Um neue Werkzeuge zu definieren, muss jeweils eine konkrete Klasse von `Tool` und `ToolView` abgeleitet werden. In der Beispielanwendung sind dies `DefaultTool` und `DefaultView`.

Im Konstruktor wird die Methode `initialize()` aufgerufen. Dort werden neue Icons und der Titeltext festgelegt, der unter dem Tool-Icon stehen soll. Auch weitere Konfigurationen oder das Wiedereinlesen gespeicherter Einstellungen sollte hier definiert werden. Im Beispiel wird von `DefaultTool` etwa das Interesse an `WobEvent` angemeldet:

```
/** Initialize the class. Put Icon and Text preferences here. */
protected void initialize() {
    super.initialize();
    Dispatcher.registerInterest(this, WobEvent.class); // !
    setText("DefaultTool");
}
```

B.3.1.1 Initialisierung

Die Icons werden schon in `InternalTool.initialize()` definiert und in `DefaultTool` übernommen:

```
/**
 * Does generic initialization work, like standard icons
 * and default popup menu listener.
 */
private void initialize() {
    // paint invisible border,
    // to avoid flicker when a green border is drawn on acceptDrop
    Color c = UIManager.getColor("Desktop.background");
    setBorder(
        javax.swing.BorderFactory.createLineBorder(c)
    );
    setBorderPainted(true);
    // Default attribute configuration
    setName("InternalTool");
    setHorizontalTextPosition(SwingConstants.CENTER);
    setVerticalTextPosition(SwingConstants.BOTTOM);
    setText("");
    setContentAreaFilled(false);
    initIcons("object0.gif", "object2.gif");
    // register listener for popup menus
    this.addActionListener(this);
    Dispatcher.registerInterest(this, WobEvent.class);
    MouseListener popupListener = new PopupListener();
    getButton().addMouseListener(popupListener);
}
```

Abgeleitete Tools werden spezifisch initialisiert. Insbesondere die Icons und der Text unter dem Tool sind von Bedeutung. Die `initIcons(String, String)` Methode erwartet die Namen zweier Bilddateien – für den geschlossenen und den offenen

Zustand eines Werkzeuges¹.

```
/**
 * Initializes the Icons for closed and open state.
 * @param ClosedIcon java.lang.String The Icon for the closed state.
 * Only leafname is needed, as the Icon will be searched on the classpath
 * (should be part of some resources jar)
 * @param OpenIcon java.lang.String as ClosedIcon
 */
public void initIcons(String ClosedIcon, String OpenIcon) {
    try {

        fieldIcon = new ImageIcon[2];
        try {
            URL location = this.getClass().getResource("/" + OpenIcon);
            setIcon(Tool.OPEN, new ImageIcon(location));
            location = getClass().getResource("/" + ClosedIcon);
            setIcon(Tool.CLOSED, new ImageIcon(location));
        } catch (NullPointerException exc) {
            handleException(exc);
        }
    } catch (Throwable t) {
        LOG.error("The Icons couldn't be initialized", t);
    }
}
```

Im HistoryTool sieht die Initialisierung dann aus, wie folgt:

```
private void initialize() {
    this.initIcons("history.gif", "history.gif");
    this.setText("History");
}
```

B.3.1.2 View erzeugen und initialisieren

Zudem muss ein neu definiertes Tool festlegen, welche Art von View es öffnet. Dafür wird die Factory-Methode `createView()` überschrieben. In `DefaultTool`:

```
/**
 * Creates a new view.
 * Initializes it with the tool as internal frame listener.
```

¹Die Grafiken werden über den Classloader gesucht – das ist wichtig, damit die Applikation WebStart-kompatibel bleibt.

```
*/  
public ToolView createView() {  
    DefaultView aView = new DefaultView(this);  
    if (aView != null) {  
        aView.addInternalFrameListener(this);  
    }  
    return aView;  
}
```

Hier kann der View auch noch gesondert initialisiert werden. In der Regel initialisieren sich die Views jedoch selbst in der eigenen `initialize()` Methode.

B.3.2 Neue Tools anmelden

Um neue Werkzeuge auf die Werkzeugleiste zu bringen, müssen sie erzeugt und dem ToolController als Argument der Methode `ToolController.addTool(Tool aTool)` übergeben werden.

Definieren aus dem Programm heraus: In der Hauptklasse `Daffodil` werden Werkzeuge in der `initTools()` Methode erzeugt und angemeldet:

```
/**  
 * Callback method for Tool initialisation.  
 * Is being called on Ancestor Event, when Desktop is shown.  
 * runs in separate Thread, for smoother display  
 */  
  
public void initTools() {  
    //fireStatusEvent(new StatusEvent(this, "Starting up tools..."));  
  
    ToolController.getInstance().addTool(getPaddleTool());  
    fireStatusEvent(new StatusEvent(this, "Personal_Lib_tool_done.));  
  
    if (personal_props != null) {  
        String startup = personal_props.getProperty("searchtool", "yes");  
        if (startup.equals("yes")) {  
            ToolController.getInstance().addTool(getSearchTool());  
            fireStatusEvent(new StatusEvent(this, "Search_Tool_done.));  
        }  
        startup = personal_props.getProperty("authornetworktool", "yes");  
        if (startup.equals("yes")) {  
            ToolController.getInstance().addTool(new ProgressTool());  
        }  
    }  
}
```

```
        fireStatusEvent(new StatusEvent(this, "Progress_Tool_done."));
    }

...
} else {
...
}
// auto open, as this Method is called in separate threas, we need to
// assure, the following is executed in the swing thread
SwingUtilities.invokeLater(new Runnable() {
    public void run() {
        getSearchTool().doClick();
    }
});
}
```

B.4 Event-Modell

Um die Austauschbarkeit bzw. die Zusammenarbeit von Komponenten aus den verschiedenen DAFFODIL-Projekten zu gewährleisten (die ggf. zum Erstellungszeitpunkt noch gar nichts voneinander wissen), soll eine gemeinsame, eventbasierte Kommunikationsstruktur aufgebaut werden. Damit kann vermieden werden, dass entweder die Module direkt voneinander wissen, oder dass sie Details der applikationsspezifischen Controller kennen müssen, von denen sie eingesetzt werden. Für jede Anwendung wird eine Hierarchie von vordefinierten Event-Klassen bereitgestellt, die die jeweiligen Ereignisse der Anwendung modelliert.

Einsatzgebiet für die hier beschriebene Event-Struktur ist weniger die Kommunikation innerhalb einer Komponente als vielmehr die Verständigung zwischen unabhängigen Modulen.

B.4.1 Kommunikation

Hauptnachteil der Java-Event-Modelle ist, dass sie es notwendig machen, dass die kommunizierenden Module einander auf Objekt-Ebene bekannt sind, oder dass sie (wie im Bean-Modell) jeweils nur mit der einen übergeordneten Komponente kommunizieren können, die sie eingesetzt hat. Für die Erstellung eines Frameworks von interagierenden aber unabhängigen Modulen, also loser Kopplung, ist aber ein generischerer Kommunikationsansatz notwendig, in dem Komponenten ihr „Interesse“ für bestimmte Arten von Ereignissen anmelden können und andere applikationsspezifische Controller ihrerseits in die Verteilung von Nachrichten eingreifen können.

Zur Umsetzung dieser Anforderungen wurde das News-Subscriber-Pattern, beschrieben in Buschmann u. a. (1996), mit dem Java-Event-Modell verbunden.

Zusätzlich wurde ein Mechanismus vorgesehen, Events in eigenen Java-Threads abarbeiten zu lassen. Damit vereinfacht sich – für einfache Fälle – die Handhabung von parallelen Threads unter Java/Swing erheblich.

B.4.2 Kommunikationsvarianten

Folgende Event-Zustellungs Methoden werden angeboten:

- **Broadcast**
Die Nachricht wird an unbekannte Empfänger versendet. Alle interessierten Abonnenten des Nachrichtentyps bekommen eine Kopie der Nachricht zuge stellt.
- **Direkt**
Der Empfänger wird direkt adressiert; nur er erhält die Nachricht.
- **postEvent**
Die Nachricht wird in die zentrale Queue eingereiht; in einem eigenen Thread werden der Reihe nach alle Events der Queue abgearbeitet; zum Bearbeitungszeitpunkt wird festgestellt, wer „interessiert“ ist. Die Funktion `postEvent()` kehrt sofort zurück.
- **sendEvent**
Die Nachricht wird sofort (im aktuellen Thread-Kontext) bearbeitet: alle Interessierten werden der Reihe nach aufgerufen, bevor `sendEvent()` zum Aufrufer zurückkehrt.
- **threadEvent**
Für jeden Interessenten wird ein neuer Thread zur Bearbeitung dieses Events gestartet. In einem neuen Thread wird die Eventbearbeitungsfunktion des angegebenen Empfängers aufgerufen. Die Funktion `threadEvent()` kehrt sofort zurück.

Es gibt folgende Methoden, „Interesse“ anzumelden:

- `registerInterest(EventReceiver view, Class cls)`
Interesse an allen Events, die von Quellobjekten der angegebenen Klasse (oder abgeleiteten) versandt werden
- `registerSrcClassInterest(EventReceiver view, Class modCls)`
Interesse an allen Events der angegebenen Klasse (abgeleitet von `java.util.Event`) oder deren Ableitungen
- `registerSrcObjInterest(EventReceiver view, Object model)`
Interesse an allen Events, die von der angegebenen konkreten Objekt-Instanz versandt werden²

²Dabei implementiert `registerInterest` das Publisher-Subscriber Pattern, während `registerSrcObjectInterest` das klassische Observer Pattern umsetzt.

Es gibt die Möglichkeit, einen Nachrichtentyp – und damit auch alle Ableitungen – abonnieren zu können. Hier wird deutlich, dass als Strukturierung für das News-Subscriber Modell die Klassenhierarchie genutzt wird: Anstelle von String-Präfixen werden Klassen (oder Interfaces) verwandt, um Gruppen von Events zu beschreiben. Damit kann man z.B. Interesse an allen „Thesaurus“ Events anmelden, oder, falls Interfaces bei der Event-Definition geschickt eingesetzt werden, an allen „Auswahl-HatGewechselt“ Events.

Beispiel:

```
class ThesaurusEvent extends java.util.Event {} ;

interface AuswahlHatGewechseltEvent implements java.util.Event {} ;

class ThesaurusMachtWasAnderes extends ThesaurusEvent{} ;

class ThesaurusAuswahlHatGewechseltEvent
    extends ThesaurusEvent implements AuswahlHatGewechseltEvent {};
```

Damit kann mit `registerInterest(ThesaurusEvent)` Interesse an allen ThesaurusEvents angemeldet werden.

`registerInterest(AuswahlHatGewechseltEvent)`
abonniert unabhängig von der Thesaurus-Hierarchie alle Auswahl-Events, auch aus anderen Modulen, die das AuswahlHatGewechselt-Interface verwenden.

Die Möglichkeit, eigene Events zu definieren, macht es einfach, beliebige eigene Daten mit dem Event weiterzureichen.

B.4.2.1 Bearbeitungsreihenfolge

Um mehr als einer Instanz zu ermöglichen, auf die empfangene Nachricht zu reagieren, wird das Pattern „Chain of Responsibility“, welches in Gamma u. a. (1995) beschrieben ist, angewandt.

Um das „Chain Of Responsibility“-Pattern zu umzusetzen wird eine Methode `registerChainParent()` implementiert, mit der für jeden Event-Empfänger ein übergeordnetes Handler-Objekt definiert werden kann, das aufgerufen wird, falls der ursprüngliche Empfänger die Nachricht nicht bearbeiten kann oder will. Hiermit kommt die zusätzliche Anforderung, dass alle Event-Handler einen booleschen

Wert zurückliefern müssen, der angibt, ob das Event erfolgreich bearbeitet wurde. Wird hier `false` zurückgeliefert, so versucht der Chain-Mechanismus, einen übergeordneten Handler zu finden.

B.4.2.2 EventReciever

In der einfachsten Ausprägung muss jede Klasse, die Events empfangen will, das Interface `EventReciever` implementieren. Dann wird in einer Funktion `handleIzevent()` die übliche Fallunterscheidung nach Event-Typen gemacht und dann die entsprechenden Funktionen aufgerufen. Ab einer gewissen Anzahl von unterschiedlichen Events führt das aber zu Unübersichtlichkeit.

Deshalb wird folgender erweiterter Mechanismus benutzt: Selbstdefinierte Event-Klassen, die das Interface `SelfDispatchable` implementieren werden nicht direkt an die `handleIzevent()` Methode übergeben - statt dessen wird ihre eigene Methode `dispatchSelf(target)` aufgerufen. Hier kann – für eine bestimmte Klassenhierarchie von eigenen Events – überprüft werden, ob das Zielobjekt ein bestimmtes problemspezifisches Interface implementiert, um dann die entsprechenden speziellen Methoden anzuspringen.

Bsp:

```
interface mySpecificInterface {
    boolean handleSpecificEvent(java.util.Event ev) ;
}

class specificEvent
    extends java.util.Event
    implements selfDispatchable
{
    boolean dispatchSelf(Object target) {
        if(target instanceof mySpecificInterface) {
            return ((mySpecificInterface)target).handleSpecificEvent(ev);
        } else {
            return ((EventReciever)target).handleEvent(ev);
        }
    }
}

class someEventReciever
    implements EventReciever, mySpecificInterface
```

```
{  
  . . .  
}
```

Damit kann für eine spezielle Bibliothek von zusammengehörigen Modulen ein übersichtliches eigenes Interface geschaffen werden.

Abbildungsverzeichnis

2.1	Iteratives Retrieval	15
2.2	Berrypicking: Eine wechselhafte Folge von Recherschritten	16
2.3	DEViD: Visueller Formalismus zur Repräsentation der Ergebnismen- gen bei Boolescher Anfragekombinatorik.	26
2.4	Elastizitätsmodul in WING-M2	27
2.5	Autorennetzwerk, berechnet über drei digitale Bibliotheken, und Hub- Berechnung, nach Mutschke (2001).	29
2.6	Brushing and Linking bei VisMeB	30
2.7	Die Google Maps Applikation	30
2.8	Prototypisches Textretrieval-Interface nach Byrd u. a. (1998).	33
3.1	Assistent für Microsoft Word, der Vorschläge zum aktuellen Kontext anbietet.	50
3.2	Der Remembrance Agent	59
3.3	Netscapes What's related	60
3.4	Wizard von Open Office	63
3.5	Automator: Automatisierte Arbeitsabläufe	65
3.6	Ein Newsaggregator wird verwendet, um automatisch die neuesten Nachrichten aus verschiedenen Quellen zu sammeln.	66
3.7	Matrix zu Systemunterstützung in Bezug auf Aktivitätsebenen.	78
4.1	Workflow-Zyklus nach Paepke	87
4.2	Verteilte Architektur von Daffodil	94
4.3	Konzeptuelle Schichtung der DAFFODIL-Dienste	95
4.4	Firefly von Maes – adaptive Recommendation.	101
4.5	Shneidermans FilmFinder – direktmanipulative Alternative zu Firefly, um interessante Medien zu finden.	102
4.6	Vorschläge aus der Query-Historie	106

Abbildungsverzeichnis

4.7	Rechtschreibprüfung	106
4.8	Overconstrained Queries	107
4.9	Vorlagen aus persönlichem oder überindividuellem Kontext	109
4.10	Extrahierte Autoren einer Resultatliste	110
4.11	Extrahierte Terme als Tag-Cloud	111
4.12	Clustering der Ergebnisse zu semantischen Gruppen	112
4.13	Daffodil Desktop mit Suchwerkzeug und Handbibliothek	116
4.14	Suchwerkzeug	118
4.15	Gewichtete Resultatliste mit Icons	120
4.16	Persönliche Handbibliothek	121
4.17	Referenzen und Zitationen	122
4.18	Koautoren-Netz Analyse Werkzeug	124
4.19	Journal- und Konferenzbrowser	126
5.1	Screenshot Mockup für die heuristische Evaluation	137
5.2	Anzahl der Anfragen je Nutzer und Aufgabe	141
5.3	Zufriedenheits-Ebene	146
B.1	WOB Desktop, Version 1	155
B.2	Klassen im Desktop	157

Literaturverzeichnis

Arms 2001

ARMS, William Y.: *Digital Libraries*. Cambridge, Mass. : MIT Press, 2001

Baeza-Yates u. Ribeiro-Neto 1999

BAEZA-YATES, Ricardo ; RIBEIRO-NETO, Berthier: *Modern Information Retrieval*. Addison Wesley, 1999

Ball u. a. 1997

BALL, Gene ; KURLANDER, David ; PUGH, David ; SKELLY, Tim ; THIEL, David ; STANKOSKY, Andy ; LING, Dan ; MILLER, John ; DANTZICH, Maarten V. ; WAX, Trace: Lifelike Computer Characters: The Persona Project at Microsoft Research. In: *Intelligent Agents*(Bradshaw 1997), S. 191–222

Bates 1989

BATES, M. J.: The design of browsing and berrypicking techniques for the online search interface. In: *Online Review* 13 (1989), Nr. 5, 407–424. <http://www.gseis.ucla.edu/faculty/bates/berrypicking.html>

Bates 1987

BATES, Marcia J.: How to use information search tactics online. In: *ONLINE* May 1987 (1987), S. 47–54

Bates 1990

BATES, Marcia J.: Where should the Person stop and the search interface start? In: *Information Processing and Management* 26 (1990), Nr. 5, 575–591. <http://www.gseis.ucla.edu/faculty/bates/searchinterface.pdf>

Belkin 1980

BELKIN, N. J.: Anomalous states of knowledge as a basis for information retrieval. In: *Canadian Journal of Information Science* 5 (1980), Mai, S. 133–143

Belkin 1996

BELKIN, Nicholas: Intelligent Information Retrieval. In: KRAUSE, Jürgen (Hrsg.) ;

HERFURTH, Matthias (Hrsg.) ; MARX, Jutta (Hrsg.): *Herausforderungen an die Informationswirtschaft: Informationsverdichtung, Informationsbewertung und Datenvisualisierung*, 1996

Belkin 2000

BELKIN, Nicholas J.: Helping People Find What They Don't Know. In: *Communications of the ACM* 43 (2000), August, Nr. 8, S. 58–61

Bergman 2001

BERGMAN, Michael K.: The Deep Web: surfacing hidden value. In: *Journal of Electronic Publishing* 7 (2001), Nr. 1. <http://dx.doi.org/10.3998/3336451.0007.104>

Borlund 2003

BORLUND, Pia: The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. In: *Information Research: an international electronic journal* 8 (2003), April, Nr. 3, S. 1–38. – <http://informationr.net/ir/8-3/paper152.html>

Bradshaw 1997

BRADSHAW, Jeffrey M.: *Software Agents*. AAAI Press / The MIT Press, 1997. – ISBN 0-262-52234-9

Brajnik u. a. 2002

BRAJNIK, Giorgio ; MIZZARO, Stefano ; TASSO, Carlo ; VENUTI, Fabio: Strategic Help in User Interfaces for Information Retrieval. In: *Journal of the American Society for Information Science and Technology* 53 (2002), Nr. 5, S. 343–358

Buschmann u. a. 1996

BUSCHMANN, F. ; MEUNIER, R. ; ROHNERT, H. ; SOMMERLAD, P. ; STAL, M. ; BUSCHMANN, Frank (Hrsg.): *Pattern-oriented Software Architecture. A System of Patterns*. 1. John Wiley & Sons, 1996

Bush 1949

BUSH, Vannevar: As we may think. In: *Atlantik Monthly* (1949). <http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm>

Byrd u. a. 1998

BYRD, Donald ; SHNEIDERMAN, Ben ; CROFT, W. B.: Sorting Out Searching: A User-Interface Framework for Text Searches. In: *Commun. ACM* 41 (1998), April, Nr. 4, 95–98. <http://www.dlib.org/dlib/january97/01contents.html>

Chalmers 1999

CHALMERS, M.: Comparing Information Access Approaches. In: *Journal of the American Society for Information Science* 50 (1999), Oktober, Nr. 12, S. 1108–1118. – Special Topic Issue: The 50th Anniversary of the Journal of the American Society for Information Science - Part 2: Paradigms, Models, and Methods of Information Science

Chojnacki u. Fischer 2005

CHOJNACKI, Michael ; FISCHER, Gudrun: Der MLHTBrowser: Interaktive Exploration von Kollektionen auf verschiedenen Abstraktionsebenen. In: *Proceedings of the 6th workshop of the GI working group "Knowledge Discovery"*, BTW2005, Karlsruhe, 2005

Curtin u. a. 1998

CURTIN, Matt ; ELLISON, Gary ; MONROE, Doug: "What's Related?" *Everything But Your Privacy*. <http://www.interhack.net/pubs/whatsrelated/>. Version: 10 1998, Abruf: 2007-12

Cypher 1991

CYPHER, A.: Programming repetitive tasks by example. In: *Proceedings CHI 91 Conference*. New York : ACM Press, 1991, S. 33–39

Dehn u. van Mulken 2000

DEHN, Doris M. ; MULKEN, Susanne van: The impact of animated interface agents: a review of empirical research. In: *Int. J. Hum.-Comput. Stud.* 52 (2000), Nr. 1, S. 1–22. <http://dx.doi.org/http://dx.doi.org/10.1006/ijhc.1999.0325>. – DOI <http://dx.doi.org/10.1006/ijhc.1999.0325>. – ISSN 1071–5819

Eibl 2000

EIBL, Maximilian: *Forschungsberichte Informationszentrum Sozialwissenschaften*. Bd. 3: *Visualisierung im Document Retrieval Theoretische und praktische zusammenführung von Softwareergonomie und Graphik-Design*. Bonn : Informationszentrum Sozialwissenschaften, 2000

Erikson u. Yeh 1985

ERIKSON, C.W. ; YEH, Y.: Allocation of Attention in the visual field. In: *Journal of Experimental Psychology: Human Perception and Performance* 11 (1985), S. 583–597

Etzioni u. Weld 1995

ETZIONI, O. ; WELD, D. S.: Intelligent Agents on the Internet: Fact, Fiction, and Forecast. In: *IEEE Expert* 10 (1995), Nr. 4, S. 44–49

Fox u. Sornil 1999

Kapitel 15. In: (Baeza-Yates u. Ribeiro-Neto 1999), S. 415–432

Friedman 1998

FRIEDMAN, Batya: User Autonomy: Who Should Control What and When? In: *ACM SIGCHI Bulletin* 30 (1998), Nr. 1, 26–29. <http://www.acm.org/sigchi/bulletin/1998.1/friedman.html>

Friedman u. Nissenbaum 1997

FRIEDMAN, Batya ; NISSENBAUM, Helen: Software Agents and User Autonomy. In: *Autonomous Agents*. Marina del Rey, CA USA : ACM, 1997

Fuhr 1990

FUHR, N.: Hypertext und Information Retrieval. In: *Hypertext und Hypermedia*. Heidelberg et al. : Springer, 1990, S. 101–111

Fuhr u. a. 2000

FUHR, N. ; GÖVERT, N. ; KLAS, C.-P.: An Agent-Based Architecture for Supporting High-Level Search Activities in Federated Digital Libraries. In: *Proceedings 3rd International Conference of Asian Digital Library*. Taejon, Korea : KAIST, 2000, S. 247–254

Fuhr u. a. 2002

FUHR, Norbert ; KLAS, Claus-Peter ; SCHAEFER, Andre ; MUTSCHKE, Peter: Dafodil: An Integrated Desktop for Supporting High-Level Search Activities in Federated Digital Libraries. In: *Research and Advanced Technology for Digital Libraries. 6th European Conference, ECDL 2002*. Heidelberg et al. : Springer, 2002, S. 597–612

Gamma u. a. 1995

GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design Patterns - Elements of Reusable Object-Oriented Software*. Reading, Massachusetts : Addison-Wesley, 1995

Gillie u. Broadbent 1989

GILLIE, T. ; BROADBENT, D.: What makes interruption disruptive? In: *Psychological Research* 11 (1989), S. 243–250

Hearst 1999

HEARST, Marti A.: User Interfaces and Visualization. In: (Baeza-Yates u. Ribeiro-Neto 1999)

Hellweg u. a. 2001

HELLWEG, Heiko ; KRAUSE, Jürgen ; MANDL, Thomas ; MARX, Jutta ; MÜLLER, Matthias N. O. ; MUTSCHKE, Peter ; STRÖGEN, Robert: *Treatment of Semantic Heterogeneity in Information Retrieval*. Bonn : IZ Sozialwissenschaften, 2001 (IZ-Arbeitsbericht; Nr. 23)

Horvitz 1999

HORVITZ, Eric: Principles of Mixed - Initiative User Interfaces. In: *CHI 99*. Pittsburgh, PA, USA : ACM, 1999, S. 159–166

Horvitz u. a. 1999

HORVITZ, Eric ; JACOBS, Andy ; HOVEL, David: Attention-Sensitive Alerting. In: *Proc. UAI 99. Conference on Uncertainty and Artificial Intelligence*, 1999, S. pp. 305–313

Horvitz u. a. 2003

HORVITZ, Eric ; KADIE, Carl ; PACK, Tim ; HOVEL, David: Models of Attention in Computing and Communication: From Principles to Applications. In: *CACM* 36 (2003), Nr. 6, S. 52–59

Ingwersen 1992

INGWERSEN, P.: *Information Retrieval Interaction*. London : Taylor Graham, 1992
<http://www.db.dk/pi/iri/>

Jordan 2005

JORDAN, Matthias: *DAFFODIL: Proaktive Vorlagefunktionen*, Universität Dortmund, FB Informatik, Diplomarbeit, 2005

Kay 1990

KAY, Alan ; LAUREL, B. (Hrsg.): *The Art of Human-Computer Interface Design*. 1990

Klas 2007

KLAS, Claus-Peter: *Daffodil - Strategische Unterstützung bei der Informationssuche in Digitalen Bibliotheken*, Universität Duisburg-Essen, Diss., 2007

Klas u. a. 2004

KLAS, Claus-Peter ; FUHR, Norbert ; SCHAEFER, André: Evaluating Strategic Support for Information Access in the DAFFODIL System. In: HEERY, Rachel (Hrsg.) ; LYON, Liz (Hrsg.): *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2004)*. Heidelberg et al. : Springer, 2004 (Lecture Notes in Computer Science)

Klas u. a. 2007

KLAS, Claus-Peter ; KRIEWEL, Sascha ; FUHR, Norbert: *An Experimental Framework for Interactive Information Retrieval and Digital Libraries Evaluation*. (Submitted for publication), 2007

Klusch 1999

KLUSCH, Matthias (Hrsg.): *Intelligent Information Agents- Agent-Based Information Discovery and Management on the Internet*. Springer, 1999

Klusch 2001

KLUSCH, Matthias: Information Agent Technology for the Internet: A Survey. In: *Elsevier Science* 36 (2001), Nr. 3

Knorz u. Kuhlen 2000

KNORZ, Gerhard (Hrsg.) ; KUHLEN, Rainer (Hrsg.): *Informationskompetenz - Basiskompetenz in der Informationsgesellschaft, Proceedings des 7. Internationalen Symposiums für Informationswissenschaft (ISI 2000), Darmstadt, 8.-10. November 2000*. Bd. 38. Hochschulverband für Informationswissenschaft, 2000 (Schriften zur Informationswissenschaft). – ISBN 3-87940-753-3

Krause 1997a

KRAUSE, J.: Visualisation, multimodality and traditional graphical user interfaces. In: *Review of Information Science* 2 (1997), Nr. 2

Krause 1997b

KRAUSE, Jürgen: Das WOB-Modell. In: *Vages Information Retrieval und graphische Benutzeroberflächen: Beispiel Werkstoffinformation*. Konstanz : Universitätsverlag, 1997, S. 59–88

Krause 1999

KRAUSE, Jürgen: Sacherschließung in virtuellen Bibliotheken. Standardisierung versus Heterogenität. In: RTZEL-BANZ, Margit (Hrsg.): *Grenzenlos in die Zukunft. Proceedings des 89. Deutschen Bibliothekartags*. Freiburg im Breisgau, 1999, S. 202–212

Krause 2006

KRAUSE, Jürgen: Shell Model, Semantic Web and Web Information Retrieval. In: HARMS, Luckhardt H.-D. I. (Hrsg.) ; GIESSEN, H. W. (Hrsg.): *Information und Sprache: Beiträge zu Informationswissenschaft, Computerlinguistik, Bibliothekswesen und verwandten Fächern, Festschrift für Harald H. Zimmermann*. München : Saur, 2006, S. 95–106

Krause 2007

KRAUSE, Jürgen: The Concepts of Semantic Heterogeneity and Ontology of the Semantic Web as a Background of the German Science Portals vascoda and sowiport. In: PRASAD, A. R. D. (Hrsg.) ; MADALLI, D. P. (Hrsg.): *International Conference on Semantic Web and Digital Libraries (ICSD 2007)*. Documentation Research and Training Centre, Indian Statistical Institute , Bangalore, India, 2007, 13-24

Krause 2008

KRAUSE, Jürgen: Semantic heterogeneity: comparing new semantic web approaches with those of digital libraries. In: *Library Review* 57 (2008), Nr. 3, S. 235–248

Kuhlen 1999

KUHLEN, Rainer: *Die Konsequenzen der Informationsassistenten. Was bedeutet informationelle Autonomie oder wie kann Vertrauen in elektronische Dienste in offenen Informationsmärkten gesichert werden?* Frankfurt : Suhrkamp-Verlag, 1999 (Suhrkamp Taschenbuch wissenschaft stw 1443)

Leung u. Aerley 1994

LEUNG, Y. K. ; AERLEY, M. D.: A review and taxonomy of distortion-oriented presentation techniques. In: *ACM Transactions on Computer-Human Interaction* 1 (1994), Nr. 2, 126–160. citeseer.ist.psu.edu/leung94review.html

Lewandowski u. Mayr 2006

LEWANDOWSKI, Dirk ; MAYR, Philipp: Exploring the academic invisible web. In: *Library Hi Tech* 24 (2006), Nr. 4, S. 529–539(11)

Lieberman 1997

LIEBERMAN, Henry: Autonomous Interface Agents. In: *CHI 97*. Atlanta, GA, USA : ACM, 1997, S. 67–74

Look 2003

LOOK, Phillip: *Daffodil: Übersicht und Entwicklung von möglichen Algorithmen für Recommendation*, Universität Dortmund, FB Informatik, Diplomarbeit, 2003

Maes 1994

MAES, Pattie: Agents that Reduce Work and Information Overload. In: *Communications of the ACM* 37 (1994), Nr. 7, S. 30–40

Mandl 2006

MANDL, Thomas: *Die automatische Bewertung der Qualität von Internet-Seiten im Information Retrieval*, Universität Hildesheim, Diss., 2006

Mann 2002

MANN, Thomas M.: *Visualization of Search Results from the World Wide Web*, Universität Konstanz, Diss., 2002. <http://www.ub.uni-konstanz.de/kops/volltexte/2002/751/>

Markoff 2000

MARKOFF, John: Microsoft Sees Software Agent as Way to Avoid Distractions. In: *New York Times* (2000), Juli. <http://www.nytimes.com/library/tech/00/07/biztech/articles/17lab.html>, Abruf: 20.2.2007

Marx 1996

MARX, Jutta: Die Computer-Talk-These in der Sprachgenerierung. Hinweise zur Gestaltung natürlichsprachlicher Zustandsanzeigen in multimodalen Informationssystemen. In: *Natural Language Processing and Speech Technology, Results of the 3rd KONVENS Conference*. Hawthorne, NY, USA : Mouton de Gruyter, 1996. – ISBN 3-11-015449-8, S. 35-42

Marx u. a. 1995

MARX, Jutta ; MUTSCHKE, Peter ; SCHOMMLER, Marcus: Möglichkeiten der intelligenten Integration heterogener Datenbestände: das Projekt GESINE / Informationszentrum Sozialwissenschaften. 1995 (Nr. 2). – Arbeitsbericht

Mayr u. Walter 2007

MAYR, P. ; WALTER, A.-K.: Zum Stand der Heterogenitätsbehandlung in vascoda: Bestandsaufnahme und Ausblick. In: *Information und Ethik 3. Leipziger Kongress für Information und Bibliothek*. Leipzig : Dinges & Frick, 2007

Mayr u. Walter 2006

MAYR, Philipp ; WALTER, Anne-Kathrin: Abdeckung und Aktualität des Suchdienstes Google Scholar. In: *Information: Wissenschaft und Praxis* 57 (2006)

van Mulken u. a. 1998

MULKEN, Susanne van ; ANDRÉ, Elisabeth ; MÜLLER, Jochen: The Persona Effect, How Substantial is it? In: *HCI '98: Proceedings of HCI on People and Computers XIII*. London, UK : Springer-Verlag, 1998. – ISBN 3-540-76261-2, S. 53-66

Mutschke 2001

MUTSCHKE, Peter: Enhancing Information Retrieval in Federated Bibliographic Data Sources Using Author Network Based Stratagems. In: *Research and Advanced Technology for Digital Libraries: 5th European Conference, ECDL 2001, Darmstadt, Ger-*

many, September 4-9, 2001; *Proceedings*. Heidelberg et al. : Springer, 2001 (Lecture Notes in Computer Science; 2163), S. 287–299

Mutschke 2004

MUTSCHKE, Peter: Autorennetzwerke: Verfahren der Netzwerkanalyse als Mehrwertdienste für Informationssysteme / InformationsZentrum Sozialwissenschaften. Bonn, April 2004 (32). – Arbeitsbericht

Müller 2001

MÜLLER, Florian: *DAFFODIL: Basisarchitektur für ein Agentensystem für digitale Bibliotheken*, Universität Dortmund, FB Informatik, Diplomarbeit, 2001

Nardi u. Zarmer 1993

NARDI, B. A. ; ZARMER, C. L.: Beyond Models and Metaphors: Visual Formalisms in User Interface Design. In: *Journal of Visual Languages and Computing* 4 (1993), S. 5–33

Negroponte 1997

Kapitel 3. In: (**Bradshaw 1997**), S. 57–66

Nielsen 1994

NIELSEN, Jacob: Heuristic evaluation. In: *Usability Inspection Methods*. New York : John Wiley and Sons, 1994

Norman 1994

NORMAN, Donald: How might People interact with Agents. In: *Communications of the ACM* 37 (1994), Nr. 7, S. 68–71

Nottelmann u. Fuhr 2001

NOTTELMANN, H. ; FUHR, N.: MIND: An architecture for multimedia information retrieval in federated digital libraries. In: *Proceedings of the DELOS-Workshop on Interoperability in Digital Libraries* DELOS-Network of Excellence on Digital Libraries, 2001

Nottelmann 2005

NOTTELMANN, Henrik: Inside PIRE: An extensible, open-source IR engine based on probabilistic logics / University of Duisburg-Essen. 2005. – Forschungsbericht

Nwana u. Ndumu 1999

NWANA, H. S. ; NDUMU, D. T.: A Perspective on Software Agents Research. In: *Knowledge Engineering Review* 14 (1999), Nr. 2, S. 1–18

O'Day u. Jeffries 1993

O'DAY, Vicki ; JEFFRIES, Robin: Orienteering in an Information Landscape: How Information Seekers get from here to there. In: *INTERCHI '93*, 1993, S. 438–445

OMG 1999

OMG: Agent Technology, Green Paper / Framingham Corporate Center. 1999. – Forschungsbericht

Ostwinkel 2002

OSTWINKEL, Stefan: *Adaptive Agenten für Daffodil.*, Universität Dortmund, FB Informatik, Diplomarbeit, 2002

Paepcke 1996

PAEPCKE, A.: Digital Libraries: Searching Is Not Enough–What We Learned On-Site. In: *D-Lib Magazine* 2 (1996), Mai, Nr. 5. – <http://www.dlib.org/dlib/may96/stanford/05paepcke.html>

Reiterer u. a. 2003

REITERER, Harald ; LIMBACH, Tobias ; MÜLLER, Frank ; KLEIN, Peter ; JETTER, Christian: Ein visueller Metadaten Browser für die explorative Erkundung großer Datenmengen. In: SZWILLUS, G. (Hrsg.) ; ZIEGLER, J. (Hrsg.): *Mensch und Computer 2003: Interaktion in Bewegung*. Stuttgart : B. G. Teubner, 2003, S. 165–176

Rhodes u. Maes 2000

RHODES, Bradley ; MAES, Pattie: Just-in-time information retrieval agents. In: *IBM Systems Journal special issue on the MIT Media Laboratory* 39 (2000), Nr. 3 and 4, pp. 685-704. <http://www.research.ibm.com/journal/sj/393/part2/rhodes.html>

Rist u. a. 1997

RIST, Thomas ; ANDRE, Elisabeth ; MULLER, Jochen: Adding Animated Presentation Agents to the Interface. In: *Intelligent User Interfaces*, 1997, 79-86

Roderfeld 2004

RODERFELD, Ingo: *DAFFODIL: Awareness auf Objekten aus digitalen Bibliotheken*, Universität Dortmund, FB Informatik, Diplomarbeit, 2004

Roppel 1995

ROPPEL, Stefan: *Visualisierung und Adaption: Techniken zur Verbesserung der Interaktion mit hierarchisch strukturierter Information.*, Konstanz, Dissertation, 1995

Scerri 2001

SCERRI, Paul: *Designing Agents for Systems with Adjustable Autonomy*. Linköping Sweden, Linköpings Universitet, Diss., December 2001. <http://www.isi.edu/~scerri/pubs.html>

Schaefer u. a. 2005

SCHAEFER, Andre ; JORDAN, Matthias ; KLAS, Claus-Peter ; FUHR, Norbert: Active Support For Query Formulation in Virtual Digital Libraries: A case study with DAFFODIL. In: RAUBER, A. (Hrsg.) ; CHRISTODOULAKIS, C. (Hrsg.) ; TJOA, A M. (Hrsg.): *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2005)*. Heidelberg et al. : Springer, 2005 (Lecture Notes in Computer Science)

Sherman u. Price 2001

SHERMAN, Chris ; PRICE, Gary: *The Invisible Web: Finding Hidden Internet Resources Search Engines Can't See*. Cyberage Books, 2001. – ISBN 0-910965-51-X

Sherman u. Price 2003

SHERMAN, Chris ; PRICE, Gary: The Invisible Web: Uncovering Sources Search Engines Can't See. In: *Library Trends: Organizing the Internet* 52 (2003), Nr. 2, 282-298. <http://hdl.handle.net/2142/8528>

Shneiderman 1994

SHNEIDERMAN, Ben: Dynamic Queries for Visual Information Seeking / University of Maryland, Department of Computer Science. 1994 (CS-TR-3022). – Forschungsbericht

Shneiderman 1997

SHNEIDERMAN, Ben: Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces. In: *1997 international conference on Intelligent user interfaces*. Orlando, FL United States, 1997, 33–39

Shneiderman u. Maes 1997

SHNEIDERMAN, Ben ; MAES, Pattie: Direct Manipulation vs Interface Agents. In: *ACM interactions* (1997), Nr. november + december, S. 42–61

Stempfhuber 2001

STEMPFHUBER, Maximilian: *Objektorientierte Dynamische Benutzungsoberflächen: ODIN. Behandlung semantischer und struktureller Heterogenität in Informationssystemen mit den Mitteln der Softwareergonomie*, Koblenz, Dissertation, 2001

Strötgen u. Kokkelink 2001

STRÖTGEN, Robert ; KOKKELINK, Stefan: Metadatenextraktion aus Internetquellen. Heterogenitätsbehandlung im Projekt CARMEN. In: SCHMIDT, Ralph (Hrsg.): 23. *Online-Tagung der DGI* Bd. 4. Frankfurt am Main : Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis, 2001 (Tagungen der Deutschen Gesellschaft für Informationswissenschaft und Informationspraxis), S. 56–66

Theobald u. Klas 2004

THEOBALD, M. ; KLAS, C.-P.: BINGO! and Daffodil: Personalized Exploration of Digital Libraries and Web Sources. In: *In 7th International Conference on Computer-Assisted Information Retrieval (RIAO 2004)*, Test, 2004

Walter u. a. 2006

WALTER, Anne-Kathrin ; MAYR, Philipp ; STEMPFHUBER, Maximilian ; BALLAY, Arne: Crosskonkordanzen als Mittel zur Heterogenitätsbehandlung in infoconnex. In: STEMPFHUBER, Maximilian (Hrsg.): *In die Zukunft publizieren: Herausforderungen an das Publizieren und die Informationsversorgung in den Wissenschaften* Bd. 11. Bonn : Informationszentrum Sozialwissenschaften, 2006 (Tagungsberichte)

Weibel u. Miller 1997

WEIBEL, S. ; MILLER, E.: A Summary of the CNI/OCLC Image Metadata Workshop. In: *D-Lib Magazine* 3 (1997), Januar, Nr. 1

Wolff 1996

WOLFF, Christian: *Schriften zur Informationswissenschaft*. Bd. 24: *Graphisches Faktenretrieval mit Liniendiagrammen. Gestaltung und Evaluierung eines experimentellen Rechercheverfahrens auf der Grundlage kognitiver Theorien der Graphenwahrnehmung*. Universitätsverlag Konstanz, 1996